S^3 DATA PROCESSING SYSTEM STUDY

Contract No. NAS5-9437

29 May 1967

PREPARED FOR

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland

Space Systems Center
Federal Systems Division
INTERNATIONAL BUSINESS MACHINES CORPORATION
Gaithersburg, Maryland

CONTENTS

CONTENTS (Cont'd)

CONTENTS (Cont'd)

CONTENTS (Cont'd)

CONTENTS (Cont'd)

TABLES

TABLES (Cont'd)

FIGURES

FIGURES (Cont'd)

FIGURES (Cont'd)

FIGURES (Cont'd)

1.0 INTRODUCTION

This report describes the design characteristics of a miniature Data Processing System for the Small Standard Satellite, summarizes some of the many design trade-offs conducted, and characterizes the kernel requirements for which the data processing system was designed. This report culminates a design study performed by IBM for the Goddard Space Flight Center under contract NAS5-9437.

1.1 Scope

The design study was based on the S^3 requirements as defined in the Request for Proposal No. 711-39255-162, the Small Standard Satellite (S^3) Feasibility Study, and in discussions between IBM and NASA during the course of the study. The proposed experiment for the first Small Standard Satellite (S^3A) was used as an example in parts of the study. Included within the study were architectural trade-offs and instruction set optimization, detail designs with different technologies to compare and select the technologies, mechanical design optimization including environmental, weight, and standardization considerations, and the detailed design and characterization of the selected system and packages.

The study involved the analysis, programming, and system sizing with different instruction sets and machine capabilities to select a set optimal for the types of functions S^3A revealed to be characteristic of S^3 missions. Since S^3A is representative of a worst case requirement, sizing the machine against this requirement yielded some measure of

what might be expected in a large configuration with respect to weight, power, volume, and reliability. At the other end of the scale, the study also included radiation testing to assure that the design will meet requirements with respect to one year immunity in the Van Allen belt, thermal analyses, and other investigations to assure practicability and proper system characterizations.

The initial configuration for the study assumed a single program design capable of operating with only one clock at a time. A major trade-off of the study involved the comparison of this single program machine, of the type originally proposed, with a multiprocessing machine wherein two programs operate concurrently and asynchronously on a clocked interrupt type of arrangement. It was concluded that either machine organization or method of operation could meet S^3 requirements for both data and time dependent sampling. However, the dual program machine could at times cause sample time jitter in the order of one-millisecond, or 1% for sample bursts as slow as 10 bursts per second, despite the fact that the basic machine cycle time is as fast as 10 μ seconds. This jitter would be due to simultaneous requests for the time consuming A to D conversions or burst processing. In S^3A where some sample bursts may occur at about thirty bursts per second, the time jitter limit of 1% would be exceeded (jitter of about 3% for these samples); if this limit holds, the single program machine or mode would be required wherein the time jitter is only as much as 10 μ seconds on occasion.

The single program machine is recommended. In the case of the dual program machine, it is possible to operate in the single program mode as may be required of the flight and to permit operation in the event of a single clock failure. It is for this reason that the two approaches are called both machines and modes. The single program machine is more reliable than the dual program machine, is less weight sensitive to increased requirements, and can also operate with either of the two clock systems.

A second trade-off involved a comparison of the characteristics of the system using a continuous tape motion with (1) fixed bit rate and tape motion, and (2) fixed packing density and slowly varying tape motion (i.e., tape motor frequency digitally-derived frequency for constant bit rate playback). While operational characteristics of the latter are more favorable, the status and pace of the tape program could not comfortably accommodate a change from a fixed tape speed.

Consideration was given to the use of three-bit bytes as compared to the use of the specified four-bit bytes. In addition, the trade-offs specified and implied in the RFP were conducted.

In the interest of clarity, and because of the very large number of alternatives considered, the report is organized in reversed chronological order. That is to say, the resultant design is treated first and the supporting trade-offs and derivation material is addressed last. All of the items specified in the RFP to be included in this report are included in a logical place within this basic outline.

This report includes the characterization of two machines: a single program machine, and a machine capable of multiprocessing with two asynchronous programs. It also includes program listings and machine sizing data for meeting the S^3A experiment requirements, and a tutorial section on programming applications. Further, it includes detailed logic designs, mechanical designs, and all of the other data inferred above, and otherwise, as necessary to permit the realistic and proper characterization of the machines and associated delta packs.

1.2 Summary

Fundamentally, the Data Processing System (DPS) centers on an architecture having a four-bit data flow; a small but powerful instruction set which includes automatic indexing and repeating in addition to multi-level indexing, branching, linkage and control; a high speed asynchronous main store for storage of both programs and data as well as index registers and a bootstrap loader; and an extensive and flexible input/output capability.

The instruction repertoire consists of but six instructions. One need not have programming experience to program the machine. The instructions are of variable length and are packed for storage efficiency. Further storage efficiency, in both size and operations, is inherent in the power of the instructions. An instruction may be fetched once but executed a number of times with an inherent repeat or indexing capability. Multiple index registers, stored in memory, and branching on index

register contents permits program sequences to be executed a number of times, and permits multi-layer nesting of programs. Moreover, the program can set a large number of discretes not only for external control, but in conjunction with branching on discretes, for internal control as well--such as for subroutine linkage. In addition, external events can cause branching to appropriate subroutines, such as for initiating a bootstrap load, for transmitting the stored program, or initiating a specified program.

Main storage cycles are requested asynchronously for program, input, and output operations and the cycle time is so short that queueing, when it occurs, is insignificant. Program and data input operations are concurrent with data output operations, and for the input and output data time bases specified--the operations are practically simultaneous. Moreover, the duty cycle of the main store is so low that the average power nearly equals standby power. The use of a single storage unit for both programs and data, rather than using a different unit for each, reduces weight, volume, and power and permits a variable allocation of main store to program and data storage in units smaller than a module.

The input/output section of the machine is generalized to provide for standard modules and permits use of 64 input channels in a variety of ways in increments of eight channels. The input/output section includes a nine-bit logarithmic compressor that can be caused by the program to provide nine-bit or eight-bit quantities (20 bits compressed to nine with 3% accuracy, or 12 bits compressed to eight with 3% accuracy), and a 10-bit analog-to-digital converter that can be caused by the program

to provide 10-bit, 8-bit, or the least significant 8 out of 10-bit conversions. It also includes message header hardware (sync pattern, frame count, satellite clock time, and frame identification tags) and the discrete output registers for internal (linkage) and external (e.g., experiment) control.

There are two clocks available in the system; a Data Synchronous Clock, wherein the clock subdivides the average rate of occurrence of some data signal into a specified number of equal intervals, and a Spacecraft Clock which operates at a specified fixed rate. Two approaches in the use of these clocks are possible: a single program approach and a dual program approach. In the single program approach either clock can be used at any one time; in the dual program approach, two different programs--one for each clock--are operated concurrently (multiprocessing) on a clock interrupt basis. In the single program approach, if some of the data is to bear a relationship to spacecraft roll or sensor look angles, the DSC system should be used. In this case, the relative time count accompanying each message can be used to determine precise time intervals for that data for which time intervals may be an important factor.

The DPS consists of three sections: memory, central control (CC), and input/output (I/O). The memory section comprises one delta pack into which a 2K word or a 4K word (4-bit byte) array can be plugged, and electronics to accommodate a 4K array. The arrays are the same size and are interchangeable. Modularity in the memory section is,

therefore, achieved with three shelf components and the selection of two to obtain 2K or 4K words of storage. Unnecessary circuits in the delta pack can be removed to save power when using a 2K array. The addressing capability would permit a 2K and a 4K module to be attached to the Central Control at the same time, although hardware for this capability was not provided.

The Central Control section, plus common parts of the I/O section, are not flight-to-flight variable. The instruction set, operation decoding, control, data flow, etc., will not change from flight to flight but should remain constant and standard. The I/O section is generalized and a near maximum configuration is provided.

In the single program machine, the Central Control together with a near maximum I/O section is packaged in three delta packs. The I/O includes: common control, message header hardware, five 8-bit discrete output registers (DOR's), 64 multiplexor channels, an A/D converter, logarithmic compressor, and six 20-bit accumulator/shifters. If additional accumulators or other functions are required between the multiplexors and experiment or vehicle systems, an additional delta pack would be needed. If less DOR's, channels, accumulators, etc., were required, the excess hardware could be removed to save power or might be used for redundancy. For S^3A, which is representative of a worst case requirement, the basic system of three logic delta packs and one memory delta pack will suffice. In fact, some hardware could be removed (i.e., some channels,

DOR's, and an accumulator). Flight-to-flight variation in the assignment of channels to analog, accumulator, and serial digital inputs is accomplished in groups of eight channels via external jumpers. External jumpers are also used for flight-to-flight variation in the assignment of discrete outputs, eight of the branch conditions, and seven of the wait conditions.

In the dual program machine, due to the added central control hardware, none of the accumulators could be accommodated in a three-delta pack standardized arrangement for the CC and I/O sections. For the same applications, the dual program machine is less reliable, will require more power, and has less adaptability or built in growth capability. The two programs operate independently, concurrently, and asynchronously, and only a segment of either can operate at any one time. Because an operating segment may at times include burst processing, program housekeeping, and format housekeeping (i.e., inserting the appropriate message header data), one program must lock out another for as long as it takes to complete its segment before enabling the clock interrupt system. While lock out time is a function of the program, times as long as one-millisecond could be expected. If a program interrupt were queued for one-millisecond, that program could not hold acquisition jitter to less than 1% for rates higher than 10 bursts per second.

As a general rule, acquiring data in bursts is to be preferred rather than acquiring it in a sequential subcommutated manner. That is to say, if each of four samples are to be acquired once per second, it is desirable to acquire all four samples in a burst once each second

rather than sequencing the samples each one-quarter second. It is desirable in terms of program storage and generally also total system response time or jitter. Burst operation and such features as simultaneous accumulation of several channels are inherent capabilities of the architecture.

For S^3A the DPS would have the following characteristics:

Single Program		Dual Program
3.24	Weight, pounds	3.66
4.6	Power, watts	5.18
4.3	Height, inches	5.3

The DPS for S^3A includes a 4K main store, 22 multiplexer channels, 5 accumulator/shifters with control gates, two 8-bit discrete output registers, logarithmic compressor, A/D converter, and message header hardware. As data acquisition is currently defined, six 256-byte sectors of main store are required for data buffering. In a program optimized for tape efficiency, the instruction byte count is 555; fewer instruction bytes could still meet requirements. Including index registers, etc., a total of 2115 bytes therefore would suffice. For this program, therefore, a 4K byte main store is required, but will provide significant growth capability. Weight, power, and height are well within the maximum system limits of 8 pounds, 9 watts, and 9 inches.

A DPS with a 2K byte storage, 16 multiplexer channels, logarithmic compressor, A/D converter, one 8-bit discrete output register, and message header hardware might be considered a minimal

system. This DPS would have the following characteristics:

Single Program		Dual Program
3.24	Weight, pounds	3.24
3.97	Power, watts	4.55
4.3	Height, inches	4.3

The weight, power, and height figures compare favorably to the 4 pounds, 4 watts, and 5 inches specified for a minimal configuration. The single program system could accommodate more channels and another DOR and still be within the 4 watt limit.

The above weight figures include MOS FET shielding considerations. The few MOS FET devices that are used are located in the center of the DPS. The test program showed that no shielding beyond that inherent in the packaging would be required for accumulated dosages of 10^{15} electrons/cm^2. The power figures represent average operating power and include a regulator for maintaining the A/D reference voltage to $\pm 0.01\%$.

For the single program S^3A configuration, the probability of no catastrophic failure in a year of operation is 0.893, and the probability of successful data return (no data loss) is 0.837. The probability of obtaining all accumulator data is 0.856, and of all analog data is 0.875. However, the probability of losing a single data point is very low; for example, the probability of a failure in an accumulator is only 0.005.

For the single program minimum S^3 configuration postulated above, the probability of no catastrophic failure in one year of operation

is 0.897, and the probability of successful data return (no data loss) is 0.872. The probability of obtaining all accumulator data is 0.892, and of all analog data is 0.881.

These are pessimistic reliability estimates, for it is still possible to operate and return data with failures in the areas considered to be of a "catastrophic" nature.

2.0 REQUIREMENTS

This section summarizes the important requirements governing the design of the DPS. This is a collation of the unaltered portions of the RFP specification, the altered portions, and the requirements brought to focus in analysis of the S^3A requirements.

The S^3 DPS is to be used for acquiring experiment and vehicle data of various types at various rates, processing this data, formating telemetry data frames, and outputting the data to a tape recorder, for subsequent dump, or to a transmitter channel. The S^3 DPS must be flexible in the acquisition and formating of data to accommodate flight-to-flight variations and changes during a given flight. Basically, it must be a stored program processor with provision for intermediate storage of data for time buffering between the variable acquisition rates and the fixed bit rate of the tape or transmitter. For full flexibility in the acquisition of data, programmed outputs must be provided for the control of experiment sensor modes.

The S^3 DPS is to be able to interface with: (1) 0 to +5 volt analog signals and convert them to 8-bit or 10-bit digital quantities to an accuracy of $\pm 1/2$ LSB and also permit reading the 8 LSB's of a 10-bit conversion; (2) 1 MHz pulses for accumulating up to a 20-bit count and converting this to an eight or nine bit quantity with an accuracy of 5% or better (counts of up to 4,000 and 1,000,000, respectively); and (3) serial or parallel digital data. A capability for as many as 64 program-addressable channels with modularity and assignability in either 8 or

16 channel increments is required. There may be a need for as many as 64 subcommutated channels with modularity of either 8 or 16 subchannels for vehicle data; this may be performed external to the DPS. A means should be provided for accelerated vehicle data acquisition during launch.

The DPS must be able to acquire data based on either the Data Synchronized Clock (DSC) or the Spacecraft Clock (SCC); an ability to acquire some data with one clock and the remainder with the other clock concurrently is desirable but not mandatory. The DSC must be able to accept synchronizing pulses in the range of 4 to 60 ppm, having pulse to pulse jitter of up to 3% and missing pulse durations of several hours, and generate from 2^2 to 2^{10} (preferably 2^{12}) pulses per average synchronization pulse interval equally spaced within 1% and at a rate of within 1% of the synchronizing pulse rate. For data to be collected on a time basis, the rate should be within about 20% of the rate specified and should be fixed to within about 1% or be derivable to 1%. Spacecraft roll rate will be controlled to within $\pm$10% of the desired roll rate.

The S^3 DPS must be able to acquire samples in bursts both continuously and on a duty cycle basis. Burst sampling may take several forms. Some analog data may be required periodically on a near-simultaneous basis such that a data group is acquired rapidly enough with respect to input rates of change that skew is not apparent.

Some accumulator data may need to be taken simultaneously for at least 80% $\pm$1% of the specified interval. In this case, accumulator inputs must be enabled for data accumulation and disabled during the

rapid sequential acquisition and compression of the accumulated data. Serial digital data samples, or any type of samples of a group which is to be collected at the same rate, may need to be collected in bursts at a specified rate, commutated in short bursts, or simply commutated, depending on the rate, the number of samples at that rate, and relationship of the samples. All of these forms may be required and each form may require different rates and data lengths. Within any portion of a data form there may be a need to alter the rates and data lengths for some specified period. Samples will also be required on a sequential continuous basis involving varying data lengths, and the sample rate - not the bit rate - will be the controlling factor. For a short period of time, data may be sampled at a very high rate followed by a longer period of relatively infrequent sampling. Thus, the DPS must have a buffer capability in order to operate with a fixed bit rate tape recorder or transmitter channel which requires continuous bit streams with message header data recurring every 1024 bit times. A total buffer capacity of 2K 4-bit bytes is specified, with consideration of construction in 1K or 2K bytes and clustering four or two together, respectively. Thus as much as 4K bytes may be desired. (Buffer and program requirements of S^3A could be accommodated with 2K bytes but with little flexibility margin.)

The amount of storage required for programs is a function of the data acquisition requirements, the method and efficiency of programming, and the power of the instruction set. The number of stored instructions specified is about 256 words of up to 4 bytes each or 1K

bytes predicated on straight line coding with acquisition instructions capable only of single channel specification. A system with more capable instructions, having 8 channel indexed data acquisition and multilevel subroutine indexing and linkage, should require less instruction storage. The DPS must have a capability of being reprogrammed by the ground via the command receiver and to read out the stored program via the transmitter for ground verification.

The tape recorder will be of an endless loop type operating continuously at either of two fixed rates, the higher rate normally used for dumping. The fixed rates will be set for each flight according to the recording requirements and desired transmission rate. When data is acquired by the DSC, therefore, the data synchronizing signal should be adjusted to within 10% of nominal to reduce tape wastage and to eliminate data loss. Any reprogramming will take into account the fact that the fixed output data rate and tape speed were selected for the previous program.

Power for the DPS will be provided to the requirements of the DPS by the spacecraft power system. Three or less voltages are desirable. Signal interface characteristics of the DPS will be accommodated by the interfacing systems in the interest of DPS standardization.

A minimal DPS configuration should not weigh more than 4 pounds, require more than 4 watts average power, nore require more than 5 inches of height in delta packs. A maximum DPS configuration, of which S^3A may be representative, should not weight more than

8 pounds, require more than 9 watts average power, nor require more than 9 inches of height.

The S^3 DPS must be able to tolerate one year exposure in the Van Allen radiation belt and tolerate launch on a Scout vehicle. The worst case radiation environment is expected to be about 10^{15} electrons/cm^2. Thermal and vibration requirements are as specified in section 3.7 of this report. The DPS should cause magnetic disturbances no greater than about 2.5 gamma at a distance of three feet.

3.0 SYSTEM DESCRIPTION

This section describes the organization; operation; logical, electrical, and mechanical design; packaging; application and programming; and other pertinent characteristics of the S^3 DPS. Inasmuch as two different machines are involved, the simple one is addressed first followed by the differences encountered in the more complex one (the multiprocessor). To avoid confusion, most of the material on the dual program machine (multiprocessor) is presented in section 4.0.

3.1 General

The S^3 DPS is a stored program processor designed for acquiring, processing, buffering, formating and outputting data and programs, and outputting and accepting control signals for flexibility in accommodating to different and changing requirements in the conduct and data recovery of experiments with standard processor hardware and programming.

The acquisition, processing, buffering, formating, control, and outputting of experiment and vehicle data is accomplished by a stored program of a small but powerful instruction set. A number of programs for different modes of operation can be stored and their execution can be initiated by signals or commands generated within the vehicle or by the ground. The DPS includes a bootstrap loader (a small fixed program) for permitting the ground to reprogram the machine via the command uplink. The ground can also cause readout of the program portion of memory via an uplink command for program verification. (Sync patterns

should be stored in sector beginnings in the program area for maintaining ground sync during readout, these must be preceded by unconditional branch instructions to jump over the sync patterns when executing programs.)

The architecture is optimized for the greatest flexibility in acquiring data and for the least amount of instruction storage and total hardware. The basic organizational philosophy is one of formating data into the buffer memory for tape and transmitter use. No formating controls are required in the output operation. This reflects the paramount requirement of data acquisition and formating flexibility for the greatest information return. The message header is also programmed into the format to be sure that the proper identification is attached and to be sure that the time count field contains the relative time of data acquisition, as any other time would be meaningless.

The DPS accommodates up to 64 addressable data channels of analog, pulse, and serial digital signal types. These types can be accommodated in modules of eight each up to a total of 64 of any combination. Five channels of serial digital data are reserved for message header data and associated hardware. One channel is required for the command uplink. The message header includes 24 bits of sync code, 8 bits of frame count for up to 256 different frames in a sequence, 12 bits of relative time to the nearest 0.005 second, and a 4-bit identification tag. There are three different identification tag registers available which, together with frame count (which is program resettable), permits con-

siderable latitude in the number of different formats usable at any one time. In addition to addressable channels, there are control and instruction provisions for accommodating subcommutating channels or subchannels.

The DPS includes analog-to-digital conversion, accumulator/shifter/compressor hardware and inputs for serial digital data. Accumulators are 20 bits long and the data is compressed under program control to floating point quantities of 8 or 9 bits to provide a 3% accuracy for 12 or 20 bit counts (counts of 4K or 1M). The A/D converter provides 8 or 10-bit digital quantities accurate to $\pm 1/2$ LSB (least significant bit) under program control; it can also, under program control, provide the 8 least significant bits (LSB's) of a 10-bit conversion for additional data compression capability to the 0.1% accuracy given that the full range is determined sufficiently often.

Additional inputs to the DPS include as many as 4 resettable bi-levels (4-bit discrete input register), a 4-bit resettable bi-level mode code, and 11 straight bi-levels. These are for branch and wait condition inputs. The source and exact function of these condition inputs is assignable by the user and are program interpretable for maximum flexibility. The mode code, 2 of the resettable bi-levels and 6 straight bi-levels are available to be used for branching to locations where desired subroutines or programs are stored for performing the desired actions. The remainder are assignable for wait conditions, such as DSC and SCC clocks for triggering program segments.

Outputs from the DPS are the fixed rate continuous bit streams of data to the transmitter or tape, and 40 bi-level discretes controlled by the program. The discretes can be operated in a steady state or pulsed mode under program control. The discretes can be individually set and individually cleared and many can be set or cleared simultaneously. There are five 8-bit discrete registers. A single command can set or clear any or all bits of a register simultaneously, set or clear the same bit positions of any of five registers simultaneously, or all 40 discretes simultaneously.

The discretes are also assignable by the user and can be used for controlling experiment modes by resetting or triggering events, and for inputting back into the DPS for use as branch conditions. In this latter case, the program can communicate to itself for performing program linkage whereby commonly used subroutines can be linked to special programs so that the subroutines need only be stored once in memory.

The DPS can acquire data at a variety of rates governed by the selected SCC or DSC clock rates. Clock rates are program selectable and sampling rates are programmable variations of these rates. The DPS is capable of sequential sampling, skipping clocks, and burst sampling to acquire data sequentially and simultaneously at a variety of different rates and packing it in buffer memory.

The clock system includes two chains: one for generating fixed rates (SCC), the other for generating fixed intervals within an apparent varying rate (DSC). Both chains are digital and derive basic timing from

a stable oscillator. Stages within each chain are available and combinations can be selected by the user for program use and interpretation. The initial stages of the fixed rate chain are used to derive clocks for machine timing, such as high speed shift pulses, basic machine cycles, and memory cycles.

The DSC chain is designed to provide 2 to 2^{12} equally spaced pulses within an integrated time interval. Basic synchronization of the DSC is derived from an external source (sun sensor, etc.) which may produce pulses having time jitters greater than three percent and may be blanked out for several hours. The DSC generates the specified number of pulses over the integrated intervals with about 0.2% pulse to pulse change of the smallest interval. It integrates the differences in rates of the external sync pulse and an internally derived sync pulse over many rolls, making corrections of about 0.2% of the smallest interval if one sync pulse persists in being faster or slower than the other. A "shadow protection" feature inhibits speed correction if three consecutive internal sync pulses occur without the occurrence of an external sync pulse until the external pulse reappears. A rapid initiation feature integrates the rate over 8 external sync pulses and divides the total by eight to achieve rapid stabilization in the face of input jitter. The internally generated sync pulse, as well as the 2 to 2^{12} equally spaced clocks, is available for input to the branch and wait condition inputs.

The operation of the system is controlled by programs stored in memory. The instruction set consists of only six instructions which have been optimized for flexibility and capability in performing the many different and varying functions required with the least amount of hardware and instruction storage.

(1) The basic data handling instruction is the Fetch instruction which addresses a multiplexer channel and specifies the number of bytes to be acquired from that channel, and in so doing, implies the processing operation to be performed. By use of the repeat index field, the same instruction can be caused to repeat the same operation a specified number of times using the same channel (for subcommutating) or incrementing through sequential channels the specified number of times (commutating or indexing). The instruction is fetched once but executed the number of times, as is or with incremented addressing, as specified. A chain bit specifies whether each execution is to be chained, i.e., executed as fast as the system can process the data, or is to be initiated on the occurrence of a single specific clock. The single specific clock to be used for non-chained instruction execution timing is user assignable. (In the two program machine, the chain bit looses its significance.) This single stored instruction can acquire as many as four bytes each from as many as eight consecutive channels.

(2) The Discrete instruction can address one or more of five discrete output registers (DOR's) for setting or clearing any combination of up to 8 bits at a time. The 40 discretes can be used for a variety of functions such

as: resetting, selecting, or stepping experiments; individually or simultaneously controlling accumulator input gates; and for attaching to the branch condition inputs of the DPS for program linkage.

(3) The Set Index instruction specifies one of 6 index registers, stored in memory, for setting initial index values of up to 64.

(4) The Branch instruction specifies one of 16 branch conditions and a branch address. One of the conditions is an unconditional branch to the specified address. Six of the conditions depend on the contents of the 6 index registers stored in memory. One of the conditions allows a ground controlled branch via the four-bit mode code to one of 16 memory locations for initiating ground specified routines. These locations can contain unconditional branches to the proper programs. The remaining 8 branch conditions can be individually specified by the user. Different clock stages, DOR positions, and other internal signals made available for this purpose, as well as external signals can be used for branching.

A Branch on index causes the specified index register to be fetched, decremented, and replaced in storage. If the resulting index value is not zero after a decrement, the branch address is used to fetch the next instruction; if it is zero, no branch occurs and the next instruction used is that starting at the next program count address. The Set index and Branch on index permits the same sequence of code to be repeated up to 64 times before branching out of the routine and, with 6 index registers, programs can be nested 6 deep.

A Branch on discrete, in conjunction with the Discrete instruction permits a sequence of code to be linked to several different sequences. For each different sequence to be linked to the same common sequence, a different discrete is needed.

(5) A variation of the Branch is the Wait instruction. The Wait instruction specifies up to 8 conditions which can be different than branch conditions to cause the DPS to idle until the specified condition is met. Used in conjunction with a clock stage, a Wait on clock instruction can be used to synchronize fetch operations with the specified clock time. Output data to the transmitter or tape is not affected by the wait instruction. In addition to programmed waiting on clocks, there is a specific clock input for use with non-chained instruction execution. The specific clock input causes instruction execution waiting without a programmed wait and can be attached to any available clock stage.

(6) The Route instruction, used to initialize modes, sets up the desired routes for data: Multiplex to Buffer to tape or Multiplex to Buffer to transmitter. It also specifies memory sectors set aside for buffering data so that the tape or transmitter will properly cycle through the frames of data placed there by the program. (In the dual program machine, two buffer areas are set; one for each program.) This instruction initializes buffer control registers which sequence the data input and output operations a frame at a time, consecutively cycling through the buffer area. If the program is operating in a frame area to which the output desires access, the previous frame will be output again so that

the output can never get ahead of the input, and frames will occur sequentially without interruption to the output.

The message output of the system is a continuous fixed rate bit stream making up continuous message frames of equal length. Message headers recur every 1024 bits. Each message header will include 24 sync bits and 24 identification bits. A fixed 24-bit synchronization code is used for receiver synchronization to the data frames. The identification code will include a 4-bit format tag to identify what format is being transmitted, an 8-bit frame count to indicate which particular frame of the given format is being transmitted, and a 12-bit satellite clock time signifying the time of the start of data acquisition for that frame.

The same program used for recording on the tape can be used for real-time transmission when the tape is commanded to dump, so that experiment data is not lost during the transmission interval and only one program is needed. Routing of data is through the buffer due to the variable input bit rates accompanying the various modes, rates, and bytes of sampled data.

If the roll rate decays when collecting data with the DSC, the fixed tape bit rate may exceed the average input bit rate. In this case, an interval of one complete frame is inserted in the output; this frame will be a repeat of the previous frame including of course the previous message header. Depending on the amount of decay, several redundant frames may be transmitted.

A single memory module is used for both program and buffer

storage in the interest of storage allocation modularity and standardization. All data and program flow in and out of memory is 4-bits wide. Data output from the memory is concurrent with data input or program output. Data output has priority over memory cycles and can interrupt an instruction fetch, for example, after two of its three bytes have been fetched. The interrupt lasts for one memory cycle after which the interrupted process continues. The memory cycle is so fast with respect to data output and input requirements that the acquisition rate is unaffected and the data output rate is constant (queueing, when it occurs, is no more than 10.4 seconds). Two memory module sizes (2K and 4K of 4-bit bytes) are obtained by a standard memory delta pack with pluggable arrays. The 2K array and 4K array actually contain 1984 and 3968 4-bit bytes, respectively, due to availability and cost considerations.

For the two program system, where each program is controlled by a different source and the programs are operated asynchronously, the architecture is altered somewhat. The SCC and DSC digital clock chains are completely parallel and operate concurrently. Inputs from these clocks cause interrupts to fetch the appropriate program where that program had left off. There are two program counters - one for each program, and two buffer areas set by the Route instruction - one for each program. Operations in each program must be controlled by wait on clock instructions and all fetching instructions are automatically command chained. Clock interrupts are disabled from the time one is encountered until a wait on clock in the executed program is encountered,

which again enables clock interrupts to gate the appropriate program counter. In addition, the output logic is altered to have one of the buffer areas take precedence over the other whenever a frame has been completed in it, while normally the other is cycled through until this happens.

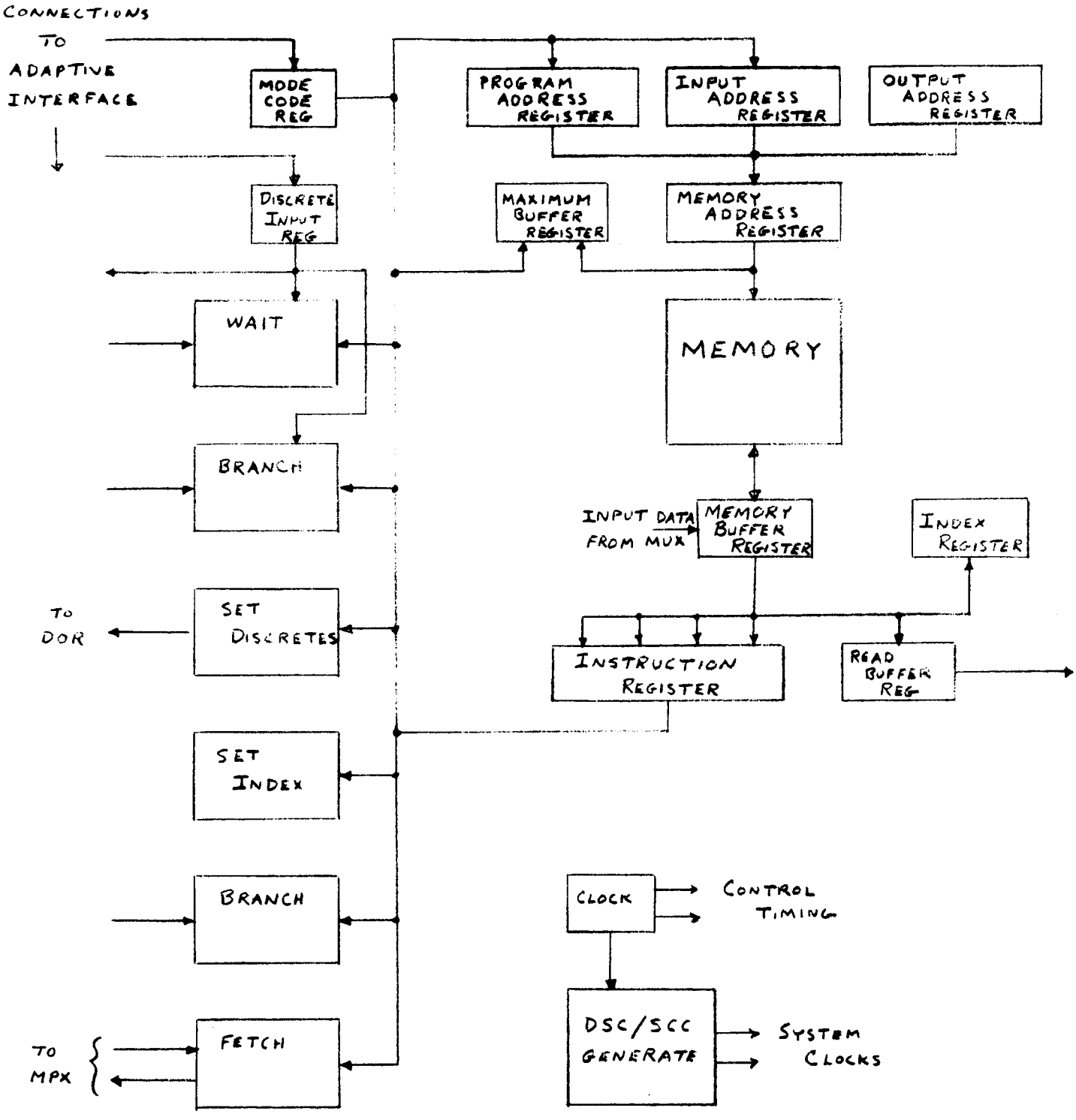
3.2 Central Control

3.2.1 General

The Central Control is that area of the Data Processing System responsible for commanding the I/O, controlling the Memory, and outputting data to the tape. All necessary timing is generated within the Central Control. Figure 3.2-1 is a simplified block diagram of the Central Control. It shows the main areas of memory addressing registers, control registers, instruction control, and clock generation each in a few blocks. The path which data takes from the I/O area to the buffer memory and back out to be output to tape is shown by heavy line.

This system is the single program system; that is, only one clock (DSC or SSC) and only one program will be used at any one time. The one program system can change clocks from DSC to SCC or vice-versa at any time during the mission that it is commanded to do so. It can execute the full instruction set and perform all the same operations with either clocking system. A two program system (described in section 4.0) has the same functional flow but with many areas in duplicate for the dual control capability.

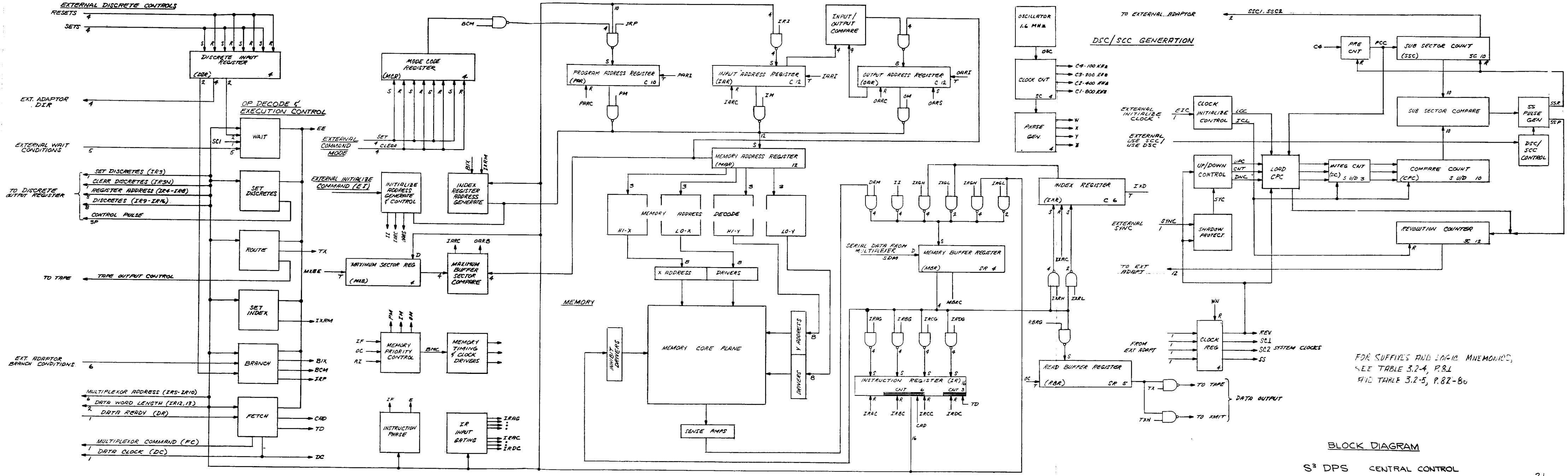
Summarizing the implementation requirements for the Central Control, it may be divided into four functional areas roughly partitioned as the logic diagrams of Figures 3.2-9, 3.2-10, 3.2-11, and 3.2-12. The following table lists the power requirements and the number of 14 pin DT^{2}L flat pack of each area.



S³ DPS CENTRAL CONTROL
SIMPLIFIED BLOCK DIAGRAM
FIGURE 3.2-1

Area	Flat Packs	Power (Watts)
Memory Addressing Register	99	0.361
Control Registers	43	0.159
Instruction Execution and Controls	71	0.227
Clock Generation	102	0.431
Total for CCU	315	1.178

Figure 3.2-2 is the detailed block diagram of the Central Control and Memory. This is the diagram which will be most useful for tracing out the data flow or instruction control as explained in the following sections.



FOR SUFFIXES AND LOGIC MNEMONICS, SEE TABLE 3.2-4, P.81 AND TABLE 3.2-5, P.82-86

BLOCK DIAGRAM

S³ DPS CENTRAL CONTROL

FIGURE 3.2-2

In the block diagram, most of the blocks are either control areas or registers. The bottom right corner of each register contains a number denoting its length (number of flip-flop stages) and often a letter: C denoting Counter; SC - Synchronous Counter; SR - Shift Register; U/D - Up/Down Counter; and S U/D - Synchronous Up/Down Counter. R and S inputs to registers are for Setting and Resetting (clearing).

The meaning of the mnemonics is explained in Section 3.2.9. A data bus is always shown as a single line but a number somewhere along its length will indicate the number of signals carried.

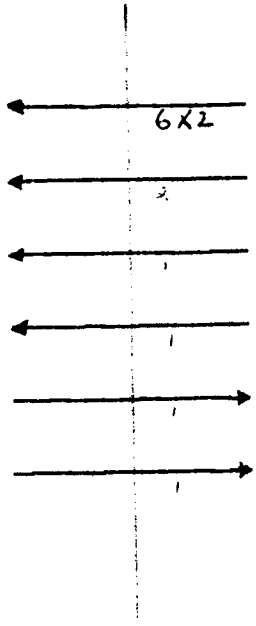
3.2.2 Data Flow

The Central Control is responsible for collecting data under program control, buffering it in the memory, and outputting it to the tape or transmitter. Data is collected by Fetch instructions which issue commands to the Multiplexor to gate in the desired data. This data is shifted serially from the Multiplexor to the Memory Buffer Register (MBR). Each byte of data, after being shifted into the MBR, is loaded into the next successive data location in the buffer memory. This multiplexing and loading of data into the buffer is under program control. Under separate hardware control is the outputting of data from the buffer to the tape. This control is based on a fixed output clock which reads out successive locations of the buffer and gates the data to the Read

Buffer Register (RBR) from whence it is serially shifted at a constant rate to the tape or transmitter.

Data from Multiplexor

The Fetch instruction controls the issuance of commands to the Multiplexor to collect data. All control and data interface lines between the Central Control and the Multiplexor are listed in Table 3.2-1. After a Fetch instruction is read into the Instruction Register (IR), it will be decoded during the Execution phase and a control signal, Command (FC), will be sent to the Multiplexor signifying that the Multiplexor Address lines and the Data Word Length lines are to be interpreted. Analog, serial digital, pulsed, and discrete data are all addressed in the same manner. The data type or form is implicit in the channel address and word length command and the multiplexor performs any necessary data conversion before sending data to the Buffer. After all data processing or conversion has been performed, the Multiplexor notifies the Central Control that the data is ready by raising the Data Ready (DR) signal. Then the Central Control generates a series of four pulses at 800KHz to serially shift each byte of data into the MBR. After a data byte is read into the Buffer, another byte is shifted to the MBR. When the number of bytes specified by the Data Word Length have been shifted to the Buffer, the Multiplexor terminates the data transfer operation by lowering the Data Ready signal. Then the Central Control is free to fetch the next instruction or re-execute the present instruction if it is a repeating or indexing Fetch instruction.



Multiplexor Address (IR5-IR10)

Data Word Length (IR12, IR13)

Command (FC)

Clock - 800KHz (DC)

Data Ready (DR)

Serial Data (SDM)

Multiplexor/Central Control Interface

Table 3.2-1

Data Into Buffer

Data from the Multiplexor is read into the Buffer during the same machine cycle as it gets into the MBR. While the data is shifting into the MBR, the contents of the Input Address Register (IAR) are gated to the Memory Address Register (MAR), decoded, and sent to the Memory. When a byte of data is in the MBR, the Memory Priority control issues a Begin Memory Cycle command to the Memory. During the following 3.2 μsec memory cycle the contents of the MBR are read into the Buffer location specified by the MAR. At this time also, the IAR, which is implemented as a simple ripple-carry counter, is incremented. Thus, the next time a byte of data is put in memory, the IAR will contain the address of the next consecutive byte in memory.

During machine initialization, the IAR will be cleared by a Route instruction so that the first byte of data will be put in the first location, $\overline{/}\,0000\ 0000\ 000\overline{0}/_2$, of memory and succeeding bytes will go to consecutive locations. The Route instruction will also load the Maximum Buffer Sector Register (MXB) with the four-bit sector address which defines the upper limit of the Buffer area of memory. This is necessary to prohibit the writing of data over the program which is already in memory. As the IAR is incremented each time data is input to memory, it is compared to the MXB to insure that it does not exceed the maximum Buffer location. When the IAR attempts to increment beyond the MXB, it is cleared to all zeros which restarts the cycle of inputting data from the beginning of memory.

Comparison of the MXB is with the four high-order bits of the IAR. These four bits of the IAR and MXB are sector addresses where a sector is defined as that amount of memory containing one frame (2^8 bytes) of locations and bounded by locations which are multiples of 2^8 (whereby the last eight bits are zero). The actual implementation of this comparison is such that the IAR sector may be equal to the maximum sector specified in the MXB but as the IAR is incremented, a carry out of the 8th position attempting to change the sector portion of the IAR will cause the entire IAR to be cleared to zero if the IAR sector is already equal to the MXB. Thus the MXB is initially loaded with the last sector address into which data is to be written.

Data from Buffer to Tape or Transmitter

Data is read out of the Buffer under hardware control. Programmable readout control is unnecessary since the data is formatted going into memory and the output data rate is constant and can be controlled by a simple clock. Data is read out to the MBR and parallel transferred to the Read Buffer Register (RBR). From here it is serially shifted out to the tape or transmitter at the Output Clock rate. Although a data word is four bits, the RBR has five stages. The extra stage is a buffer to insure a stable, exactly clocked output bit rate regardless of any delay (up to 10 μ sec) which might be experienced in retrieving each successive

byte of data from memory. One bit of the Route instruction is used to set a control which routes the output data from the RBR to either the tape or transmitter. This feature precludes any loss of experiment data while the tape is being dumped by enabling the direct transmission of data from the Buffer.

Memory is addressed for output data every fourth Output Bit Clock (OBC). The OBC is counted down by four and a signal requests memory for output. The Memory Priority area will honor this request at the completion of the present machine cycle. A machine cycle time is 10 μ sec so this is the longest possible delay in retrieving output data. This interruption of the normal program sequence for output data may occur during the Instruction Fetch phase (I-Fetch) when only part of an instruction has been read from memory or during the Execution phase when some operation is being carried out. Since the output data request is immediately honored, the program execution may experience a "jitter" of 10 μ sec at relatively infrequent times.

When the Memory Priority controls issues the output data command, the contents of the Output Address Register (OAR) are gated to the MAR and a Begin Memory Cycle command is sent to the Memory. While the data is being read out of memory to the MBR, the OAR is being incremented so that the next time it is used, it will address the next Buffer Memory location. The OAR, just as the IAR, is compared with the MXB

which contains the maximum data Buffer address. When the incremented OAR exceeds the MXB value, the OAR is cleared to all zeros and the next output data byte will be read from the beginning of memory.

The Output Clock rate is selected about 10% higher than the nominal average input rate to permit $\pm$ 10% variation about the nominal roll rate. This is to insure that all data collected at the specified rates will be outputted to tape and transmitted to ground. However, because of the output rate being slightly higher than the input rate, the output address will tend to catch up with and occasionally pass the input address. If this were allowed to happen, output frames of data would occasionally contain the ID code and some data of one frame and also some data remaining from a previously collected frame. To preclude this possibility, an Input/Output Comparator compares the OAR with the IAR whenever the OAR finishes with one sector (a frame of data) and attempts to go to the next. If the IAR indicates that data is being input to that sector, the OAR will not be allowed to advance. Rather, it will be reset to the beginning of the previous sector and that frame of data will be reread. After repeating the complete frame, the OAR will again try to advance the output address to the next sector by the same process.

During machine initialization, the OAR will be set to all one's which is the highest address in memory. In attempting to increment to the next location (all zero's because of the wrap-around addressing) the Input/Output Comparator will find that the IAR is inputting data into that

sector and the OAR will be forced to read the last sector (beginning with location 1111 0000 0000). The data readout will not be experiment data but this feature insures that the valid data output will begin soon after the first sector of memory has been filled.

3.2.3 Control Phases

There are two instruction phases; the I-fetch phase during which all bytes of the next instruction are read from Memory to the Instruction Register (IR) and Execute during which the operation specified in the IR is carried out. The I-fetch is a burst operation, using the memory continuously until the full instruction is in the IR. At the beginning of an I-fetch the Program Address Register (PAR) contents are transferred to the MAR and a Begin Memory Cycle command is sent to Memory. The addressed memory byte goes to the MBR and then is transferred to the first byte location in the IR. While this is taking place, the PAR is being incremented. Partial decoding of the first byte of an instruction indicates whether the instruction is 2, 3, or 4 bytes in length. Then the remaining bytes of the instruction are gated into successive byte locations of the IR by repeating the process until the I-fetch is completed. The PAR contains only the ten low-order bits of memory address. The additional two address bits which are necessary to address a 4K memory will always be forced to one's when the PAR is used thereby insuring that the program area will always be in the high 1024 bytes of Memory.

The Execution phase is different for every instruction as will be discussed in Section 3.4.1.5. This phase may be short or long. An instruction like Set Discretes will be executed immediately and the following instruction read without missing a machine cycle. The Wait instruction, on the other hand, hangs up the machine doing nothing in the Execution phase until the Wait condition is satisfied. Both the Execution and the I-fetch can be interrupted by an output data request but this would add only 10 μ sec (one machine cycle) to the I-fetch time and generally wouldn't interfere at all with the Execution time.

3.2.4 Instruction Set

The instruction set for the single program DPS is listed in Table 3.2-2. Execution times, including instruction read time, are given in Table 3.2-3 as minimum, maximum, and fixed times. The following pages contain an architectural description of each instruction. A simplified characterization of each instruction is given in the following thumbnail sketch:

Route - The initialization instruction. Sets up data paths and defines Buffer area.

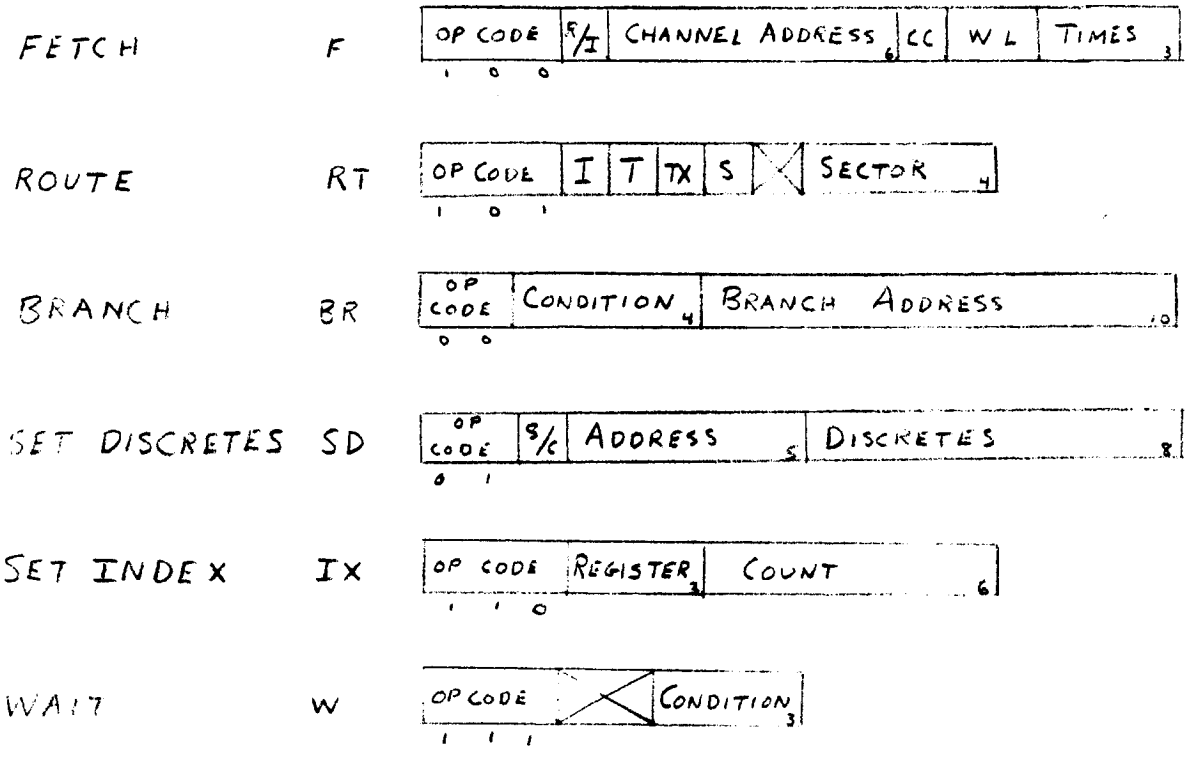
Fetch - Addresses and stores all data. One Fetch may be executed up to eight times with the same or incremented address.

Branch - Unconditional or 15 conditions including 6 index registers.

Set Index - Load one of 6 index registers. Maximum count of 63.

Set Discretes - Up to 40 discretes individually set and cleared communicate with experiments and other sub-systems, and for program linkage.

Wait - Puts machine in non-active state until specified condition is satisfied.



Instruction Set

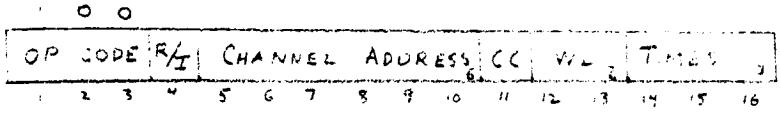
Table 3.2-2

Table 3.2-3

Instruction Read and Execute Time

	Minimum Time (μ sec)	Fixed Time (μ sec)	Maximum Time (μ sec)
Fetch	50		Depends on I/O Section, type of data, number of bytes, and whether Fetch is repeated or incremented.
Route		30	
Branch	40		80
Set Discretes		40	
Set Index		50	
Wait	20		Depends on occurrence of the specified condition

FETCH (F)

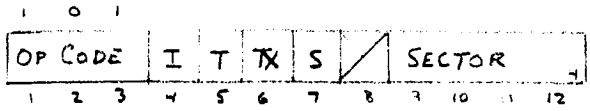


The Fetch instruction addresses the multiplexer to select data to be gated to the buffer. The six-bit Channel Address is sent directly to the multiplexer to specify the data to be selected. The two-bit Word Length is also sent to the multiplexer; this specifies if the selected data should be 1, 2, 3, or 4 bytes in length.

The three Times bits indicate the number of times that the instruction execution is to be repeated. If the times bits are zero, the instruction execution will not be repeated. The execution may be repeated up to seven times for a total of eight executions. If the R/I bit is a zero (Repeat), the same Channel Address and Word Length bits are sent to the multiplexer each time the instruction is repeated. If the R/I bit is a one (Increment), the instruction will be repeated with the Channel Address incremented by one each time.

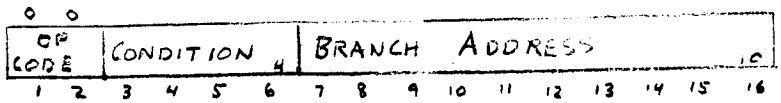
When the Command Chain (CC) bit is set to a zero, the Fetch instruction is not executed immediately, but waits for a specific data acquisition clock (the SS Clock) before addressing the multiplexer. The source of the SS Clock is user specified. The Command Chain (CC) bit set to a one, however, indicates that this instruction should be executed immediately. Command chaining, particularly with repeating or incrementing, is used to gather hi-speed bursts of data.

ROUTE (RT)



This instruction is used to initialize the Central Control by setting up the data flow routes and defining the memory area available for data buffering. When the I bit is set to one, it causes the initialization of the Input and Output Address Registers. This process forces the Input Address Register to all zeros which is the beginning of memory. The Output Address Register is forced to all ones, the highest address. Data will initially be output from the highest sector of memory until the input data has filled up one complete sector. The Tape bit (T) can be used to control the tape recorder for recording data from the buffer or reading data to the transmitter. While a ground command may be preferable for this function, this instruction would be used when it is desired to begin data transmission only at the completion of data sampling cycle. Data read from the buffer may be routed to either the tape recorder or transmitter. The route is set up by the TX bit. When the S bit is one, the 4 bit Sector field is loaded into the high order bits of the Input Address Register (IAR). When it is a zero, the Sector field goes into the Maximum Buffer Register (MXB) to define the upper limit of the data buffer and thereby prevent the writing of data over the program. Setting the IAR and MXB separately permits sectors to be skipped for inputting data to preclude data loss due to failures in that sector. It also permits the program area to be read out for ground verification of the program without inputting data over the program area.

BRANCH (BR)



The Branch instruction executes a simple branch if the condition specified is not satisfied. In branching, the ten bit Branch Address field is gated into the Program Address Register. In a conditional branch, if the condition is satisfied, the instruction stored after the branch instruction is fetched; i.e., it is a branch out of the loop on the specified condition.

Conditions 0 through 5 refer to index counters. When specified, the six-bit index word is read out of memory, placed in the Index Register, and decremented. The decremented index count is returned to its proper location in memory. If the count is not zero, the branch is executed. When the index count becomes zero, the condition is satisfied and the next stored instruction is fetched.

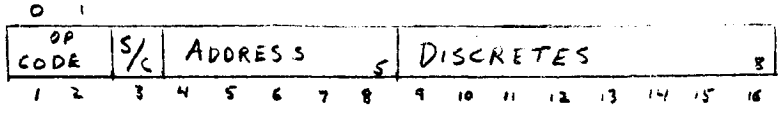
Condition 6 refers to the Mode Code Register (MCR). If this register contains all zeros, no branch is executed. If the MCR is non-zero, the MCR will be logically ANDed with bits 3, 4, 5, and 6 of the branch address. This address will be loaded into the Program Address Register. This gives a 16 way branch based on a ground command in the MCR.

Condition 7 is an unconditional branch to the specified address.

Conditions 8 and 9 are dependent upon the state of the first and second bits of the Discrete Input Register. This register may be set by inputs from ground control, on-board experiments, or on-board control and status monitors.

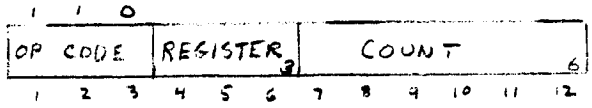
Conditions 10 through 15 are user connected to monitor whatever conditions may be of interest. These will be useful when connected to clocks, the Discrete Output Register, satellite mode indicators, experiment status monitors, and ground command registers.

SET DISCRETES (SD)



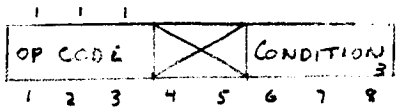
The Discrete instruction is used to turn on or off the individual stages of the specified Discrete Output Register. There can be as many as five eight-bit discrete registers for controlling internal and external events. The S/C bit specifies whether the selected register positions are to be turned on or off. The Address bits specify which register is to be acted upon and the eight Discrete bits indicate which bits of that register are to be set or cleared. Discretes can be individually set or cleared, and as many as eight in one group, or register, can be set or cleared simultaneously. Also, the same bit positions in any or all of the registers can be set or cleared simultaneously. Thus, all 40 discretes can be set or cleared simultaneously with one instruction.

SET INDEX (IX)



This instruction merely initializes the contents of the index register specified by the Register bits. There are six index registers. The index register specified, which is a location in memory, is loaded with the Count field. This specified index word will be fetched and decremented when a Branch instruction is conditional upon the index count of that index register.

WAIT (W)



The Wait instruction causes the program being executed to halt exectuion until the condition specified by the Condition bits is satisfied. While the machine is in the wait state, data transfer out of the buffer to the tape or transmitter will not be inhibited.

Condition 0 is System Clock 1 which may be user connected to any one of the clock stages available.

Conditions 1 and 2 are dependent upon the state of the third and fourth bits of the Discrete Input Register. This register is set by inputs from ground control, on-board experiments, or on-board control and status monitors.

Conditions 3 through 7 are user connected to monitor whatever conditions may be of interest. This flexibility is similar to the user defined conditions available with the Branch instruction. Often these conditions will be connected to various timing signals to cause program execution and data collection on specific clock times. There are twelve DSC clock stages and two SCC clock stages made available for use as branch conditions.

3.2.5 Instruction Execution

The preceding section presented the architecture of the DPS as is necessary for the programmer to use the system capabilities. This section discussed the details of the implementation and execution of each instruction, including the design considerations which went into the implementation of the capabilities of this system. Sufficient detail is provided to use this section as an explanation of the detail logic diagrams, Section 3.2.9.

The I-fetch and Execute control phases were explained in Section 3.2.3. The current phase is indicated by a flip-flop with complementary outputs IF and E. During the I-fetch, the Instruction Byte Counter (IBC) (see Figure 3.2-11) gates each instruction byte into its proper location in the IR. Decoding the IBC along with the instruction op code determines when the complete instruction is in the IR. This automatically terminates the I-fetch and puts the processor in the Execute phase. Every instruction, after completion of its execution, will generate an End Execute (EE) signal which will switch the machine back to the I-fetch phase.

Fetch

Decoding the Fetch instruction at the beginning of execution will set the Fetch Command (FC) flip-flop. This is the command signal which indicates to the I/O Section that the Channel Address lines and

the Data Word Length lines are valid commands. The FC signal remains up and when the I/O has selected the addressed data and performed whatever conversions or processing is necessary, the Data Ready (DR) signal from the I/O will initiate the actual data transfer. The Central Control then generates a four bit Data Clock (DC) at 800 KHz to serially shift the addressed I/O data into the Memory Buffer Register. At the same time the IAR is gated into the MAR and then the new data is written into Memory. This process continues until the I/O drops the DR line signifying the operation is complete and causing the FC signal to drop. The Fetch op decoder then examines the 3-bit Times field of the instruction to see if the execution is to be repeated. When the Times field is not zero, the Times field is decremented, the Channel Address field is incremented (if the instruction is specified as incrementing), and the Fetch Command is reissued to the I/O by setting the FC flip-flop. The execution of a Fetch is complete only after the Times field has reached zero. Then the End Execute (EE) signal puts the machine in the I-fetch to get the next instruction.

Wait

The implementation of the Wait instruction is quite simple. There is nothing for this instruction to do and no control signals for it to generate until the Wait Condition is satisfied and the EE signal is issued. The machine just remains in the Execute phase but doing nothing

(except, perhaps, for an occasional independently controlled data output) until the EE appears. There are eight Wait Conditions, three of which are internally wired with the five remaining being brought out to the connector for user adaptation to mission requirements.

Set Discretes

This instruction is used only to control the setting or clearing of bits in the Discrete Output Register. Its execution consists of issuing one control pulse to the I/O to inform it that the five Register Address lines and the eight Discrete lines are valid at that time. Immediately following this an EE is generated and the execution is complete.

Route

The Route instruction is another short instruction in execution time. At the beginning of the Execute phase, the op code is decoded and the required routing flip-flops are set. The Sector field of this instruction may be loaded into either the MXB or the IAR conditional upon the 7th bit of the instruction. Immediately following this, the EE is generated to begin the next I-fetch.

Set Index

The Set Index instruction loads an initial count into one of six index registers which are located in memory. The Count field contains

the value to be loaded into the location specified by the Register field. Execution of this instruction requires two memory cycles - one for loading the first four and one for loading the last two bits of the Count field into memory. The index registers are located at successive two-byte locations beginning with location 3962. At the beginning of Execution, the desired index register address is forced into the MAR. This address will be 3962 plus the register indicating bits of the instruction. Binarily, this is 1111 0111 $R_1R_2R_3$ 0 where $R_1R_2R_3$ are the register bits. After the first four bits of the new count are loaded into this location, the lowest bit of the MAR is forced to a one and the rest of the new count is written into the following location. At the end of the second memory cycle, the index register load is completed and the EE signal is generated.

Branch

There are four major forms of this Branch instruction: a conditional branch based on a single specified condition, a sixteen way branch based on a four-bit register, a branch which is conditional on an index register, and an unconditional branch. The basic purpose of this instruction is to put the Branch Address into the Program Address Register unless the condition specified is satisfied.

Of the 16 conditions, one is for the mode code register, six are for the six index registers, and one is for the unconditional branch. Of the eight remaining conditions, two are hard wired to the Discrete

Input Register, stages 1 and 2. The remaining six conditions are available at the connector for user adaptation to mission requirements. In cases where the specified condition is satisfied, the branch should not be executed, so an immediate EE is generated terminating the operation. If no EE is generated, the branch will be executed and an EE will then return the machine to the I-fetch before a machine cycle has been lost.

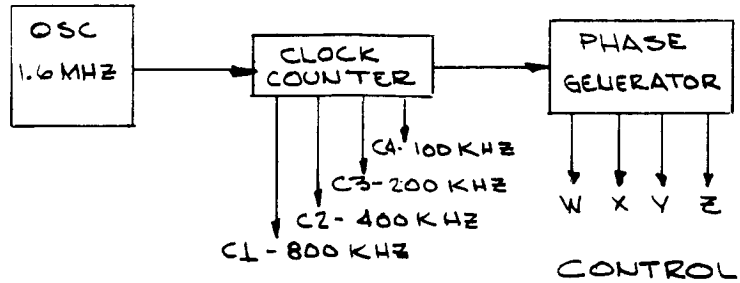
The Branch on Mode Code Register (MCR) is both a conditional branch and a multiple way branch. If the MCR contains all zeros, EE is generated and no branch is executed. Otherwise, a branch is executed with the four bits of the MCR being logically ANDed with bits 3, 4, 5, and 6 of the Branch Address Field. If the Branch Address field had been all ones, then the final branch address bits 3 through 6 would be the contents of the MCR and a 16-way branch would be executed. However, the programmer has the capability of setting any of these bits (3 through 6 of the Branch Address field) to zero and thus mask out any of the MCR bits. In this way, the branch can be conditional upon any desired combination of the MCR bits. Following loading of the new branch address into the PAR, EE is executed.

The execution of a Branch on Index instruction generally requires four memory cycles after the instruction has been read. First the specified index register must be addressed and read out of memory into the Index

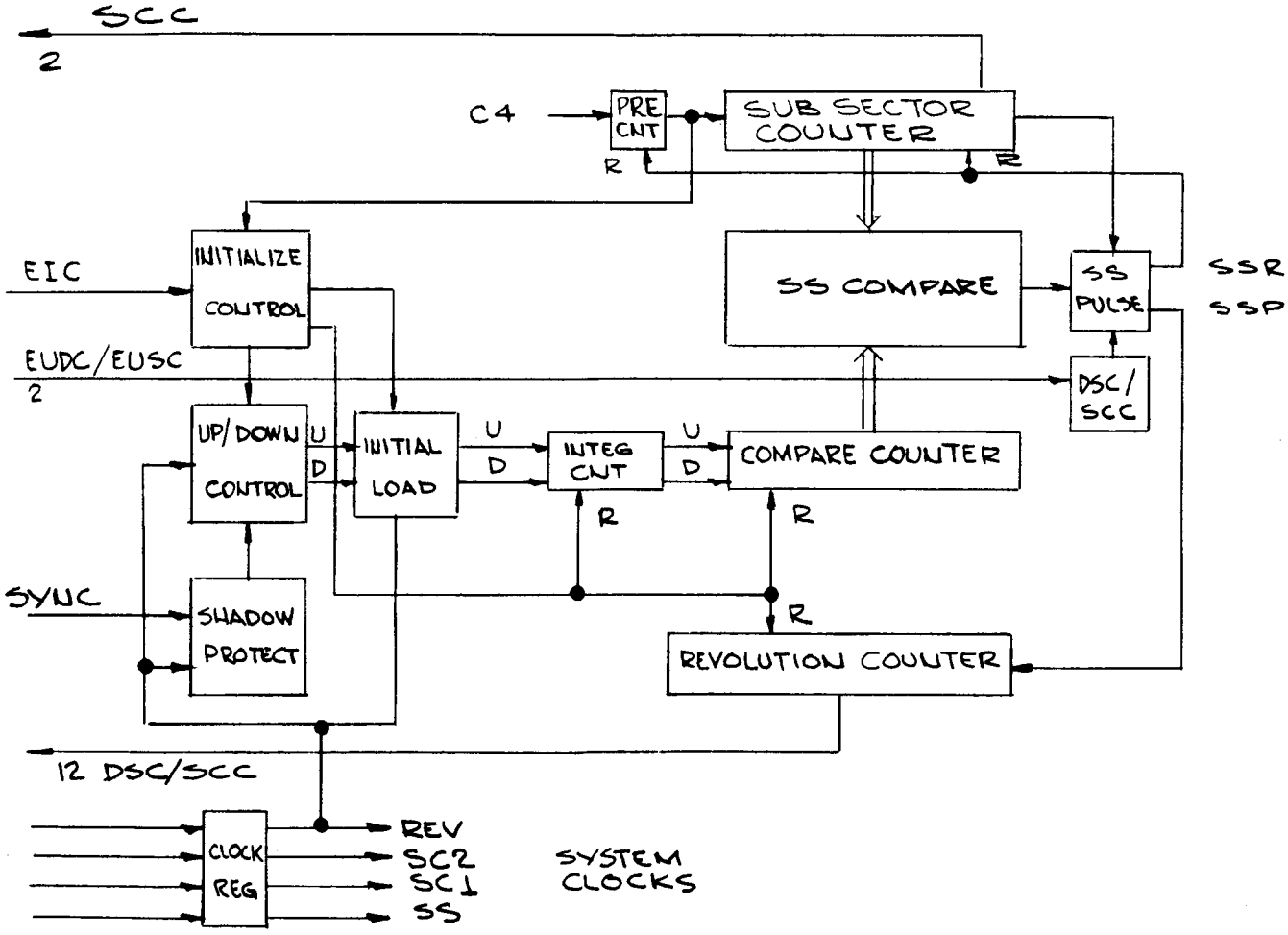
Register. The memory address is generated by the same control used in the Set Index instruction. Reading an index register takes two memory cycles, each byte being buffered in the MBR and then gated into the IXR. When the IXR contains the index word, it is decremented by one. If the result is zero, no branch is to be executed and an EE is generated. If the IXR is non-zero, the Branch Address is to be loaded into the PAR and also, the decremented contents of the IXR must be returned to that index register's location in memory. This requires another two memory cycles and is carried out in much the same way as a Set Index instruction except that the new count comes from the IXR.

3.2.6 Clock Generation

There are two general areas of the clock system. The first is the relatively high frequency area used to generate the control timing such as machine cycle time, phase clocks, and high speed shifting pulses. The second area generates the system clocks used for data collection and experiment timing. The timing for this area is derived from the control timing. The system clocks generated will be either the fixed frequency Space Craft Clock (SCC) or the externally synchronized Data Synchronous Clock (DSC). A block diagram of the entire clock generation system is shown in Figure 3.2-3.



CONTROL TIMING



CLOCK GENERATION

FIGURE 3.2-3

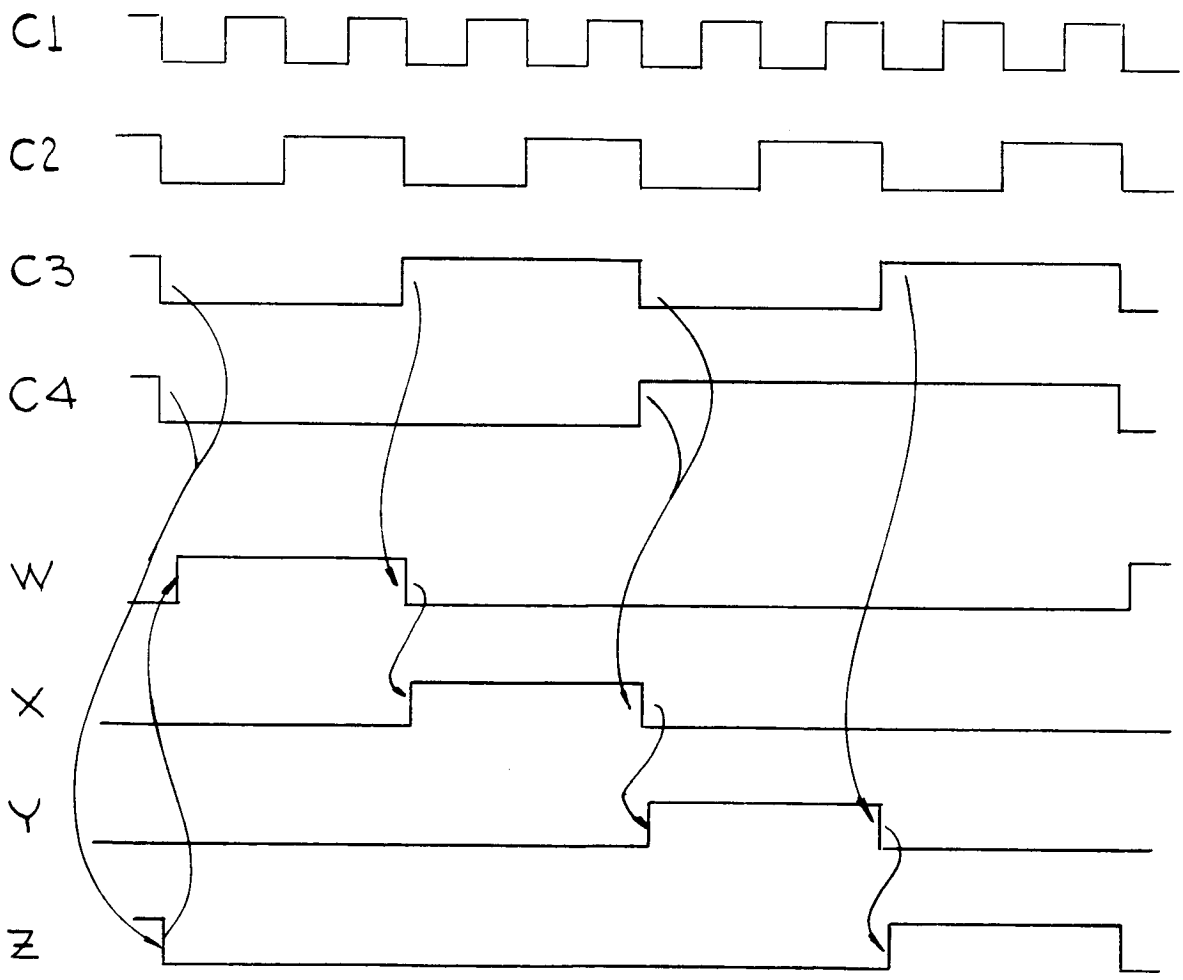
Control Timing

The output of the Oscillator is a 1.6 MHz square wave. This signal is fed into the Clock Counter, a four stage synchronous counter which generates the four high speed control clocks (C1 through C4) at frequencies of 800 KHz, 400 KHz, 200 KHz, and 100 KHz. C4 is the 100 KHz clock having 10μ sec per cycle which is the basic machine cycle time for the DPS. There is no more than one memory cycle per machine cycle. A machine cycle may be considered as the basic time required to carry out any one-byte operation, such as reading an instruction byte, storing a data byte, or reading an output data byte.

C3 and C4 are used as timing for the Phase Generator which supplies the most highly used clocks for the system. These clocks (W, X, Y, and Z) are 2.5μ sec non-overlapping pulses each of which occurs once per machine cycle. Figure 3.2.4 shows a timing chart of the high speed clocks and phase clocks. Included in this diagram are arrows indication how the changing of the high speed clocks will turn off a phase clock. To guarantee that the phase clocks are non-overlapping, any phase clock can only be turned on by the previous stage changing to the off state.

Space Craft Clock

The Space Craft Clock (SCC) is a set of fixed frequency clocks



CONTROL TIMING - HIGH SPEED CLOCKS & PHASE CLOCKS

FIGURE 3.2-4

derived from a straight-forward countdown from the control timing clock, C4. A ground command, External Use Space Craft Clock (EUSC), will cause the clocking system to generate the SCC, and since the SCC and the DSC have a great deal of circuitry in common, only the SCC will be generated.

C4 is fed into the Pre-Counter and Sub-Sector Counter (SSC). The SSC output is fed into the Revolution Counter. That puts a total of 23 stages of binary counters in series dividing down the 100 KHz clock, C4. The last 14 stages of this counter chain are user adaptable for providing system clocks from about 100 Hz, which should be highest required clocked sampling rate, to 0.715 cycles per minute. 0.715 cpm is one cycle per 84 seconds which should fulfill the long time sampling requirements although further programmed counting down from this rate is easily accomplished.

The 14 available clocks of the SCC are all of fixed frequency relative to each other and their absolute rate cannot be varied within a mission. However, the oscillator circuitry could be varied from mission to mission over a range of 2 so that it is possible to produce any frequency desired exactly (within the tolerance of the oscillator circuitry) if a particular frequency is necessary. Varying the oscillator frequency would cause a proportional variation in the control timing and thus in the basic machine cycle time. The only obvious effect that this would have on the system operation would be that program interference caused by the

Output Clock stealing a machine cycle for output data would be increased from 10 μ sec to 20 μ sec.

Data Synchronous Clock

The Data Synchronous Clock (DSC) is generated to produce as many as 4096 equally spaced pulses during the period between the arrival of externally generated sync pulses. Should the sync pulses be lost for any period of time, the DSC will continue generating clocks without any change in rate. If a correction is made to the DSC rate to compensate for a long term drift in the Sync rate, the actual change in clock to clock spacing at the time of the correction will be about 0.2%. Variation or jitter in the arrival of the externally generated sync signal of up to $\pm$ 50% from each clock's nominal time will not effect the DSC rate. The random loss of sync pulses will cause a DSC variation from true frequency of only about 0.025% per sync pulse dropped.

The basic philosophy of the DSC generation is to internally generate sync pulses and compare them with the external sync pulses. If the internally generated pulses are more frequent or less frequent than the externally generated sync pulses, then an up/down counter will tend to drift in such a way as to compensate for the difference in pulse rates. This up/down counter is the basis for the DSC generation since every time a fixed rate counter matches its contents, a sub-sector pulse

is generated; as the contents of the up/down counter change, the pulse rate changes proportionally.

The details of the DSC generation are more easily followed referring to its block diagram, Figure 3.2-3. The Sub-Sector Counter (SSC) is the high speed counter which is compared to the up/down counter, Compare Counter (CC). The 100 KHz control timing clock, C4, is fed into the 11 stage Sub-Sector Counter chain (including a one stage Pre Count). Every time the SSC compares with the CC, a Sub-Sector Pulse (SSP) is generated and the SSC is reset to zero so it can start to count all over again. The lowest possible frequency of the SSP would be about 50 Hz. Normally, a system will be set up with the SSP pulse in the area about 100 Hz to allow for full use of the counter without chancing overflow.

The SSP feeds into the Revolution Counter (RC), one stage of which is considered to be the internally generated sync pulse and connected to the Clock Register REV position. The stage of the RC which is used to form the REV signal is user determined to fit the requirements of each mission. Thus, the internally generated sync pulse (REV) is a countdown from the Sub-Sector Pulse (SSP) so that there will be 2^n SSP's per REV, where n may be set from 0 to 12.

The REV signal rate is compared with the sync pulse rate to determine any correction necessary to the DSC rate. This comparison is simply an up/down counter with each sync pulse causing an up count and

and each REV pulse causing a down count. The up/down counter is 13 stages; three in the Integrating Counter and ten in the Compare Counter. The Integrating Counter (IC) serves the purpose of smoothing out irregularities in the sync pulse. The Compare Counter (CC) contains the actual count which is compared to the Sub Sector Counter (SSC). It is noted that while the SSC is a high speed counter, the CC contains a relatively constant value with a rate of change less than one count per 800 external sync signals when the DSC is within 1% of exact synchronization over this interval.

A shadow protection feature enables the DSC to maintain a constant rate if the sync pulses disappear altogether. Whenever this control receives three REV pulses in a row without having received an external sync pulse, it assumes that the external sync pulse generator is temporarily out of commission, so it adds a count of two to the Integrating Counter to compensate for the extra REV pulses which caused down counts and then all inputs or corrections to the up/down counter are inhibited until the reoccurrence of the externally generated sync pulse. The extra pulses which get added in and subtracted from the up/down counter will generally only effect the Integrating Counter and not introduce a change in the DSC rate. This is a part of the necessity for the IC.

Initialization of the DSC is accomplished by integrating the Compare Counter loading clocks over eight revolutions (during eight sync

pulse periods) to minimize the effect of sync pulse jitter in establishing an initial value for the CC. Assuming the worst case sync pulse variation of $\pm$ 3% from nominal, the CC will be loaded to generate the DSC to within 0.375% accuracy on initialization. The initialization process is somewhat the reverse of normal operation. In normal operation some value, K, is stored in the Compare Counter. Every time the Sub-Sector Counter (SSC) counts up to K of the Pre Count (PC) pulses, a SSP is generated to increment the Revolution Counter (RC), the n^{th} stage of which generates the REV signal. Thus, in the time between sync pulses, there are K x 2^n of the PC pulses. To initialize, the CC must be loaded with a K which is generated by dividing the PC pulses by 2^n simply by feeding PC pulses into the RC and using the REV signal. Since initialization is over eight sync periods, the REV signals cannot be used directly to load the CC but must be counted down by eight in the three stage Integrating Counter. At the end of initialization, the comparison value K is in the CC, normal operation paths are restored, the RC is cleared to zero, and the IC is set to four $(100)_2$ which is half way between an up or down carryout to the CC.

System Clocks

The 12 stages of the Revolution Counter and the upper two stages of the Sub Sector Counter may all be used as system clocks.

All 14 of these clocks are connected to an external adapter so that any may be connected back into the DPS wherever required. Four clocks can be wired back to the Clock Register (CR) which provides system clocks. The first stage of the CR is for use of the clock, labelled SS, upon which the execution of a Fetch instruction waits when it is not in the command chain mode. The SS clock need not be wired to some arbitrarily defined sub sector of a revolution but rather to that clock which is most useful to the programmer in collecting data. This would generally be the highest frequency clock used for timing data sampling. The fourth stage of the clock register (REV) is wired to the RC stage which should coincide with the sync pulse in rate. This stage is internally connected for DSC initialization and for rate comparison with the sync pulse for DSC rate compensation. SC1 and SC2 are other System Clocks which may be used as desired within the DPS and are connected as conditions for the Wait instruction.

3.2.7 Initialization and Program Load

Initialization is a two step process. First the clock is initialized (if the DSC is to be used) as described in Section 3.2.6 and then the Central Control is initialized and the program started. The Central Control initialization involves setting up a bootstrap load program in memory, clearing and setting various control and data registers, and loading the Program Address Register.

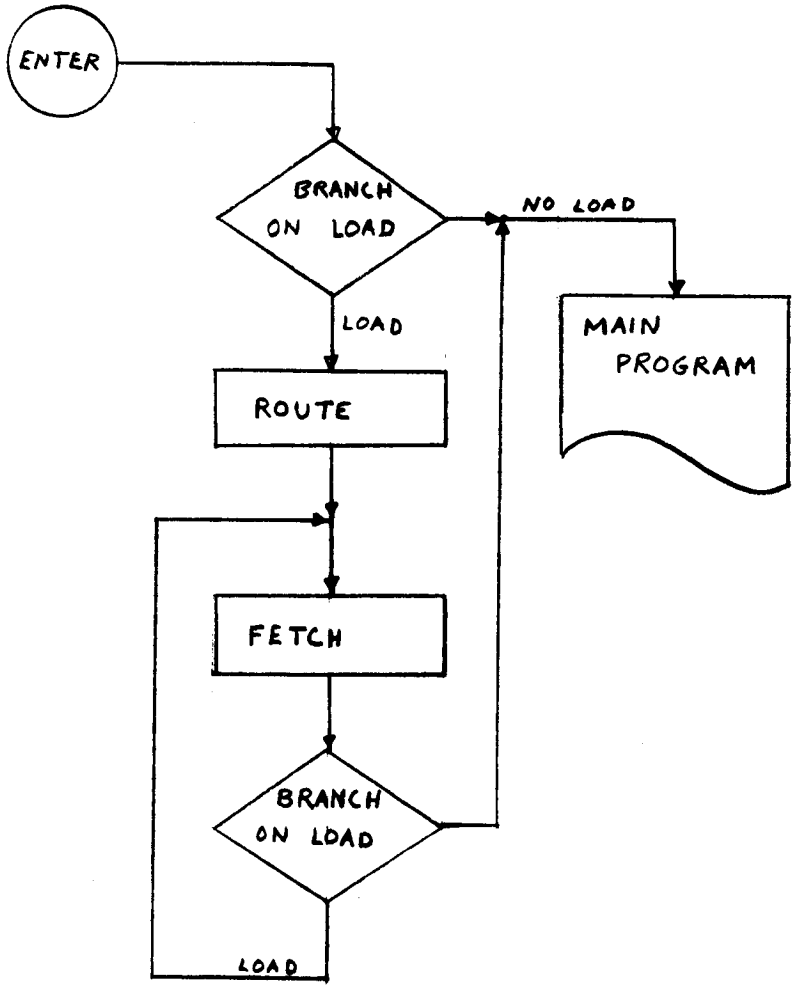
A load program is stored in memory which is protected from destruction although it does require initialization. This program contains four instructions or 16 bytes in memory with the bit locations which contain zeros having their cores broken away. Thus, the load program is essentially wired in since the broken core locations will always be zero. The bit locations containing ones are normal core locations and these must be initialized by attempting to write ones in all bit positions of the 16 byte load program. The hard wired initialize control insures a valid load program by setting ones in the core locations of this program. Then the PAR is set to the starting address of the load program with the machine in the I-fetch phase.

The first instruction of the load program is a conditional branch depending on the state of a discrete input called the load discrete. Depending on the conditions of this discrete, the program goes to the location of the main program, or the load routine. The load routine consists of a Route instruction to set up data paths, and initialize the input address register, a Fetch to read new program data into Memory, and a Branch to go back and repeat the Fetch until notified by the condition of the load discrete that the program load is complete. The data for a program load is sent via the command input hardware of the I/O section, and a normal Fetch instruction addresses it just as though it were experiment data.

In fact, the only difference between loading a program into Memory and the normal buffering of data is that the program is loaded into the upper part of memory and no experiment data will be allowed in this area. Programs can be loaded at a rate of 40K bits per second.

The lowest possible program location is 3072 (1024 in a 2 K memory system). This is where a program load always begins. The program will always be packed into the uppermost part of the memory. If the program requirements are such that only the last sector of memory (beginning at location 3840) is required, then the program load subroutine may load nonsense data sent up by telemetry in the locations from the beginning of program load (3072) to where the actual program is to begin (3840). The program load will continue, loading valid program instructions from location 3840 to the end of the program requirements.

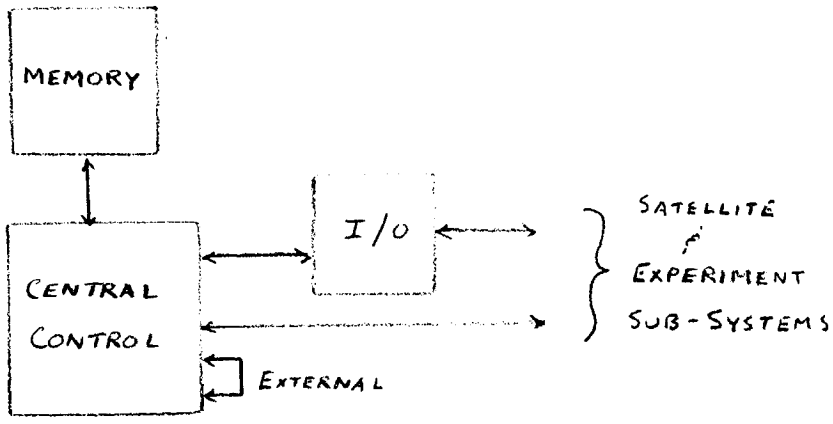
Initialization will be necessary every time the DPS is turned on but the program load routine will be used only when it is desired to load a new program (or initial program) to adapt the system to a changing experiment or subsystem conditions. To load a new program when the machine is operating, the ground must set the independent initialize discrete and the load discrete. The former forces the machine to the beginning of the load program, the latter causes the branch into the actual load routine. The four instruction load program is shown in the following flow diagram:



LOAD PROGRAM

3.2.8 Central Control Interfaces

The Central Control has interfaces with three areas; Memory, I/O, and External (see figure below). All interfaces with the Memory will be via connector plugs since the Memory and Central



Central Control Interfaces

Figure 3.2-5

Control are packages in separate delta packs. The External interfaces are positions on a connector plug which may be connected to subsystems

external to the DPS (e.g., tape recorder) or interconnected with other plug positions to give the DPS individual characteristics to match each mission's requirements. Functional interfaces with the I/O are not all through connectors since mechanical considerations necessitate packaging some of the Central Control with the I/O. Partitioning of the Central Control among delta packs, likewise, requires the use of connector pins which are not connecting functional interfaces. This section lists only the functional interfaces between the Central Control and other areas.

Memory Interface

The signals of the Central Control/Memory Interface are listed in the following table. These signals are self explanatory.

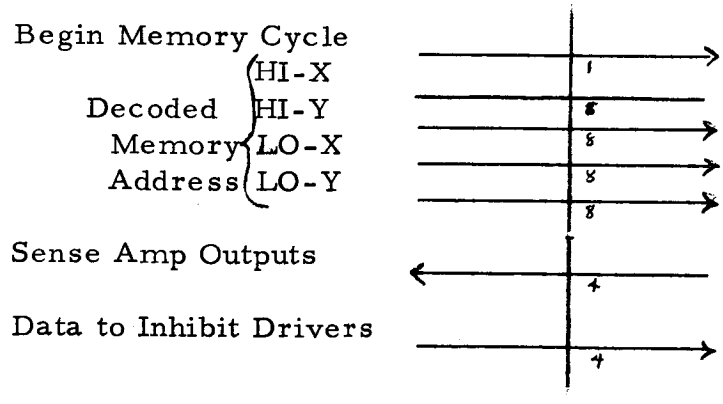


Figure 3.2-6

Central Control/Memory Interface

I/O Interface

The Central Control interface with the I/O has two areas, the Discrete Output Register (DOR) control and the data multiplexing control. These interfaces are listed in the table below:

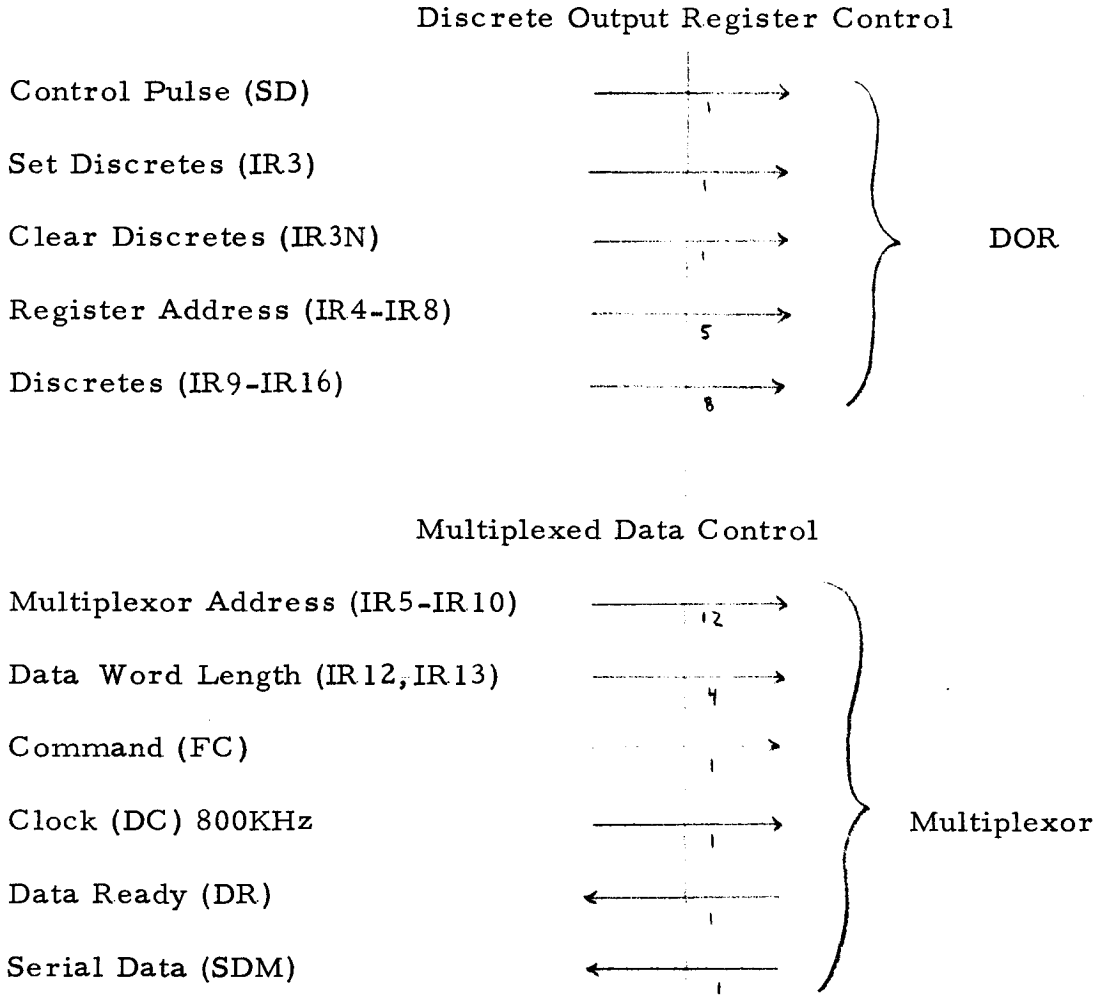


Figure 3.2-7

Central Control - I/O Interface

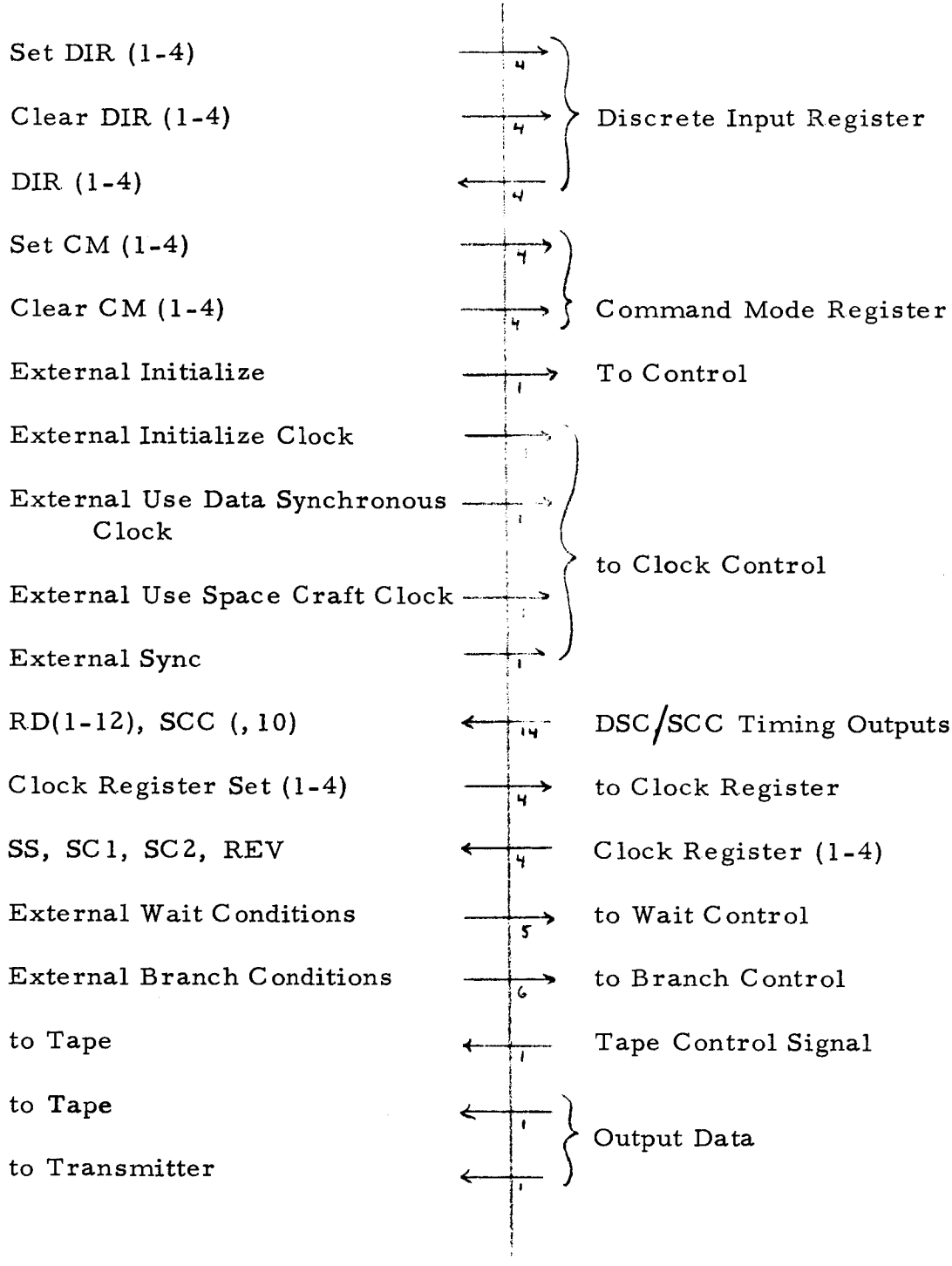
External and Adaptive Interface

The External and Adaptive Interface is used for interfacing signals between the Central Control and external systems and for adaptive wiring back into the Central Control, or to the I/O section, of those signals necessary to give each DPS its required mission characteristics. The signals of this interface are listed in Figure 3.2-8 below. Signals such as the Discrete Outputs and Multiplexor control inputs are also adaptive but are treated in the I/O section. The adaptive interface connections wired back to the DPS may be considered as personality connections giving a DPS the individual timing and control capabilities required by its particular mission data sampling rates, spin rate, and control requirements.

Adaptive interfaces include the capability for user specification of many of the branch and wait conditions, user selection of clock stages and their usage, and user selection of discrete input register sources. After selection of the primary adaptive interfaces for a mission, unused interfaces should be considered for alternate or back-up modes.

Figure 3.2-8

External and Adaptive Interface

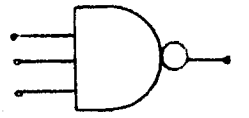


3.2.9 Detail Logic Design

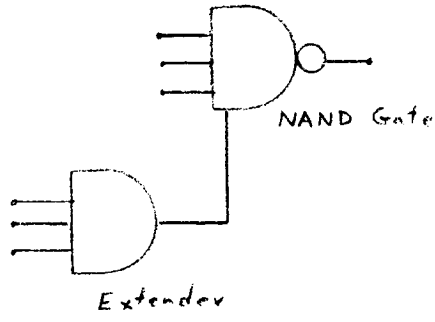
Figures 3.2-9, 3.2-10, 3.2-11, and 3.2-12 contain the detailed logic design for the Central Control. This is the detailed design of the all parts of the block diagram shown in Figure 3.2-2 except the Memory area. The circuits used here are all of the Low Power DTμL family except for a few high fan-out drivers. The gates perform the NAND function. Where the outputs of gates are dotted together, the positive AND function is generated. AND input extenders are used also. The flip-flop has several modes of operation; synchronous J-K and R-S and asynchronous direct operation. Extensive usage is made of all modes of operation.

Symbols

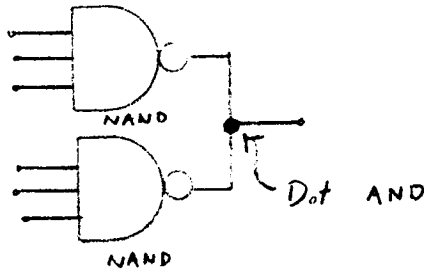
The logic symbols in Figures 3.2-9 through 3.2-12 are individually explained below. The basic logic gate is the NAND gate:



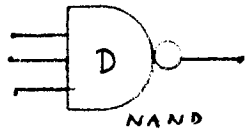
The inputs to a NAND gate may be increased using an AND extender:



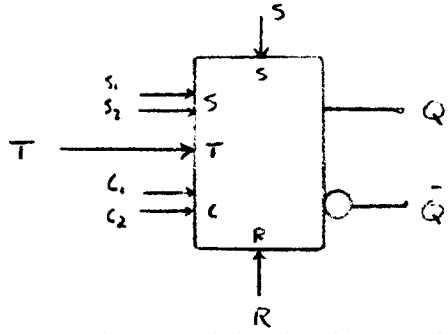
The outputs of some NAND gates may be dotted together forming the positive AND function:



The driver circuit is a regular DTμL circuit with a fan-out to LPDTμL of about 80. It performs a NAND function.



The flip-flop has triggered and direct sets and clears. The direct clear is shown as input R (for reset) to avoid some confusion. The triggered sets and clears are each 2 input NAND gates.



This explains how the DTμL is used logically in the following diagrams. For the complete detail characterization of this family of logic, refer to the Fairchild LPDTμL literature.

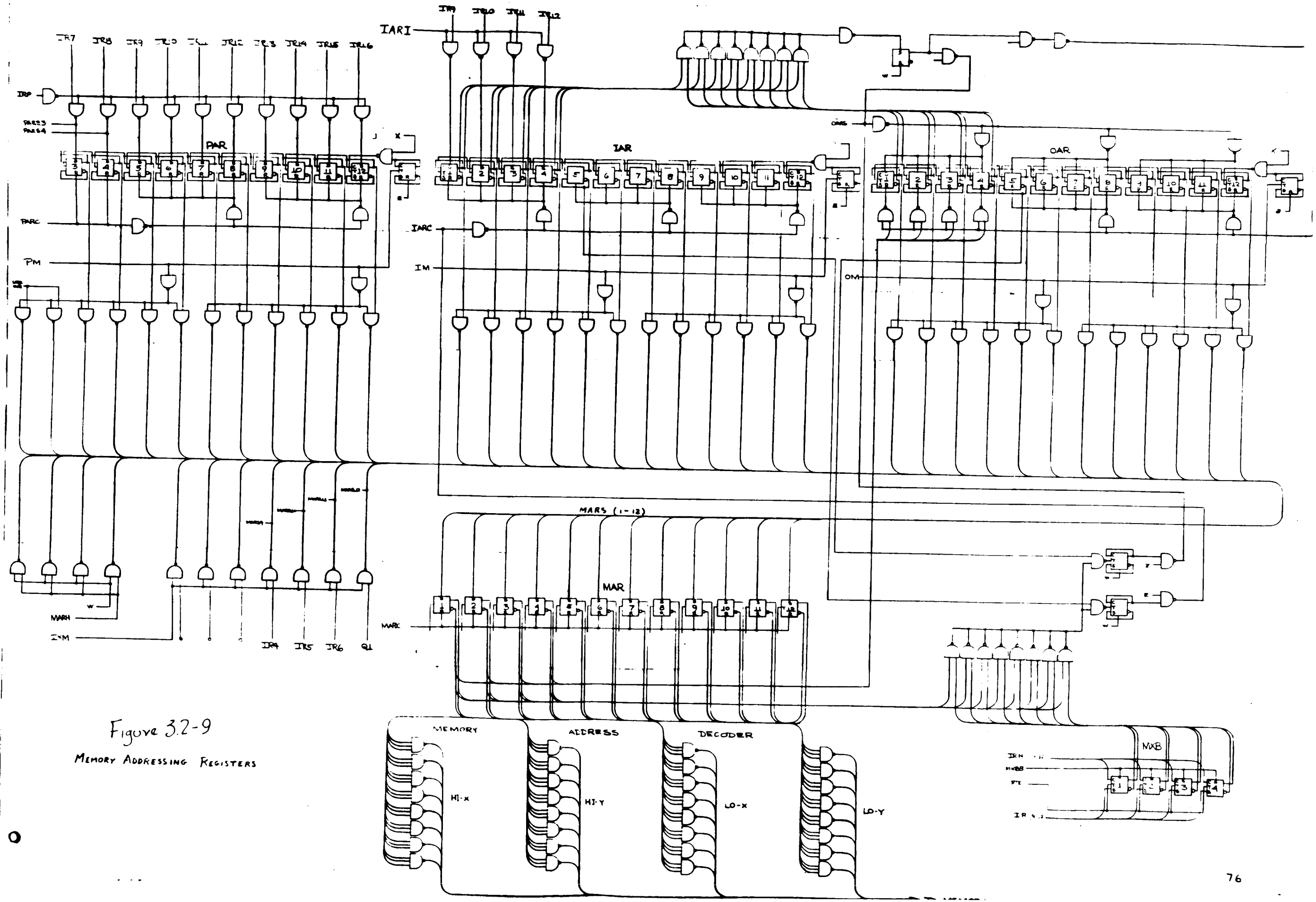


Figure 3.2-9

MEMORY ADDRESSING REGISTERS

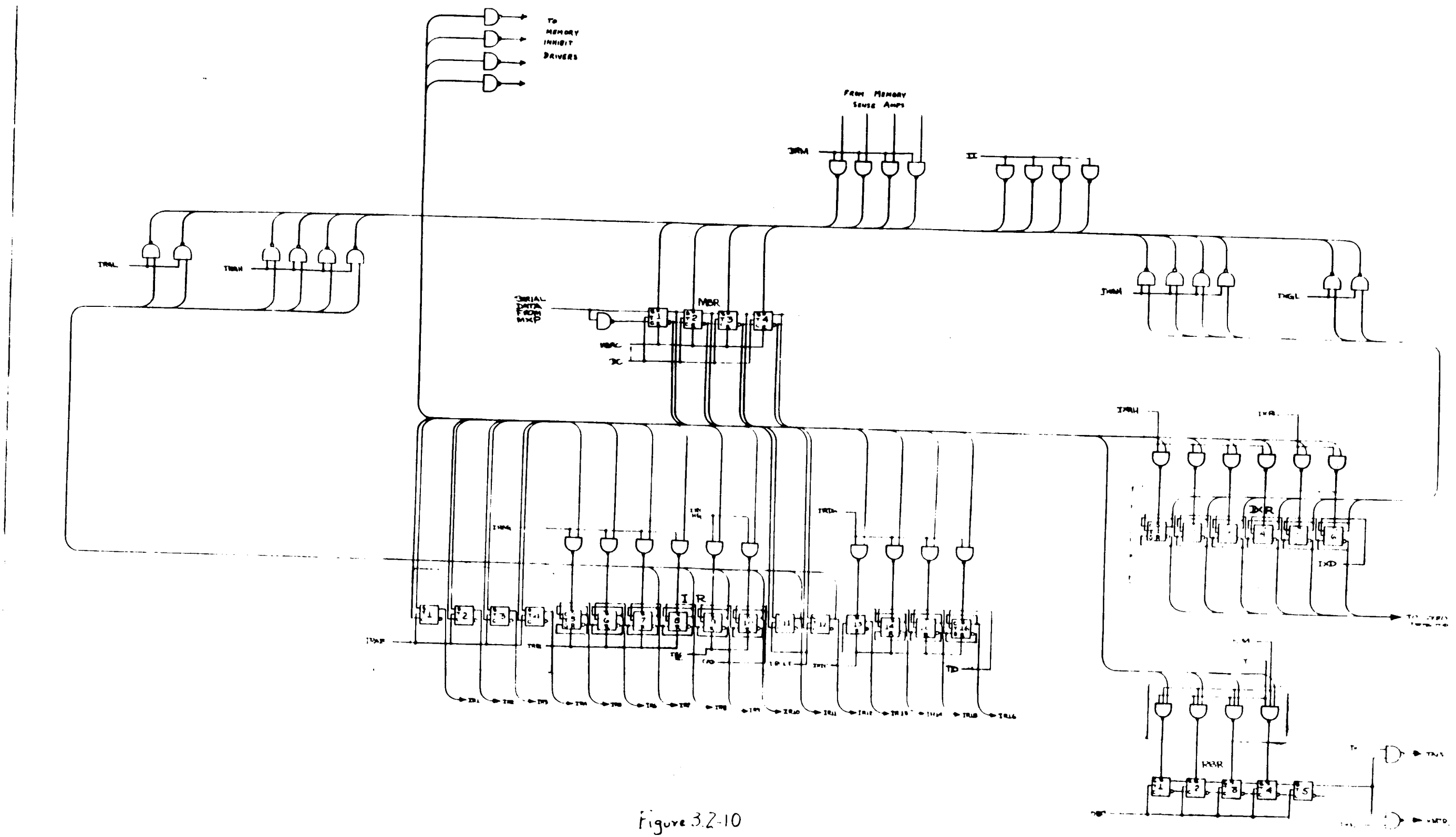


Figure 3.2-10
CONTROL REGISTERS

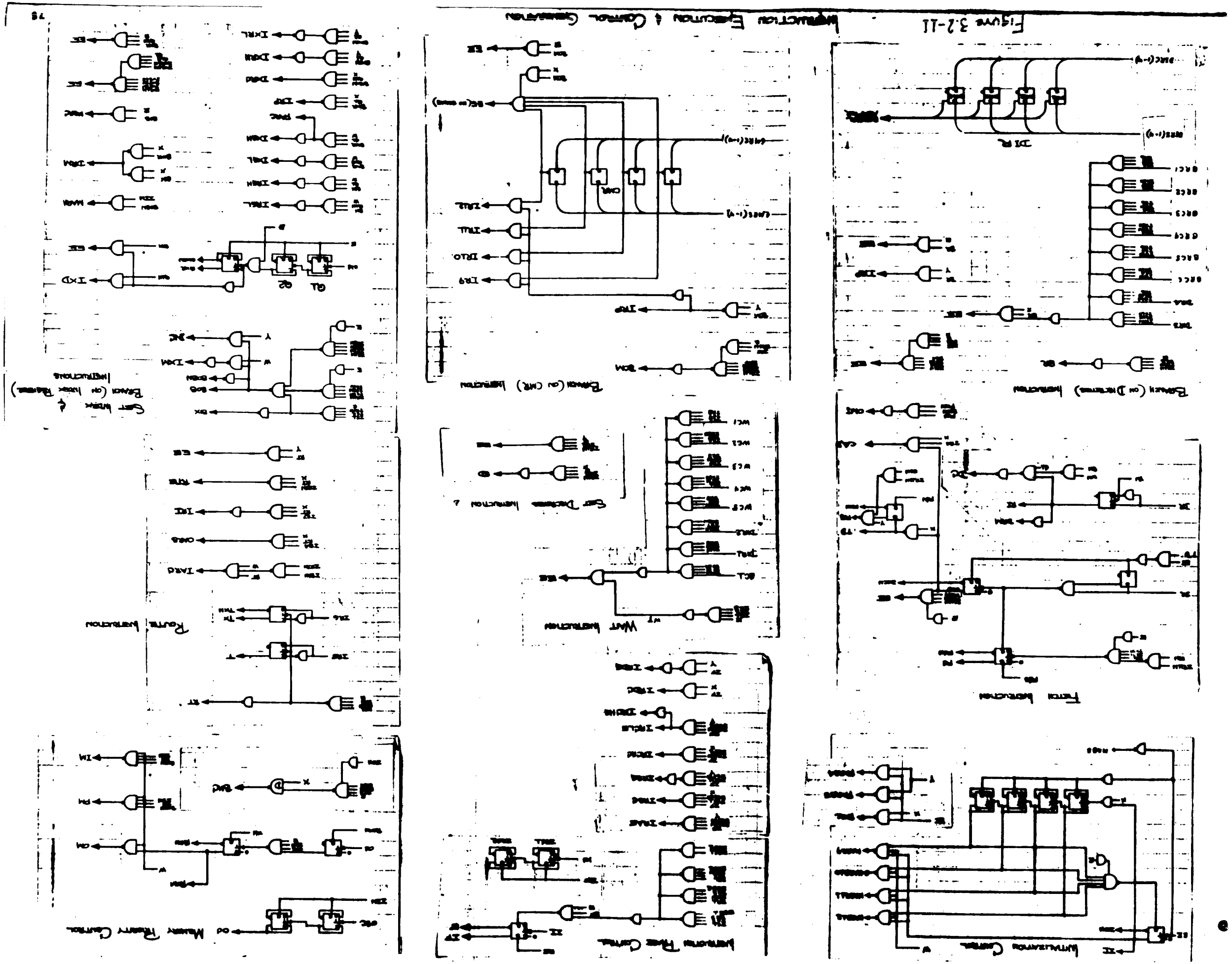


Figure 3.2-11

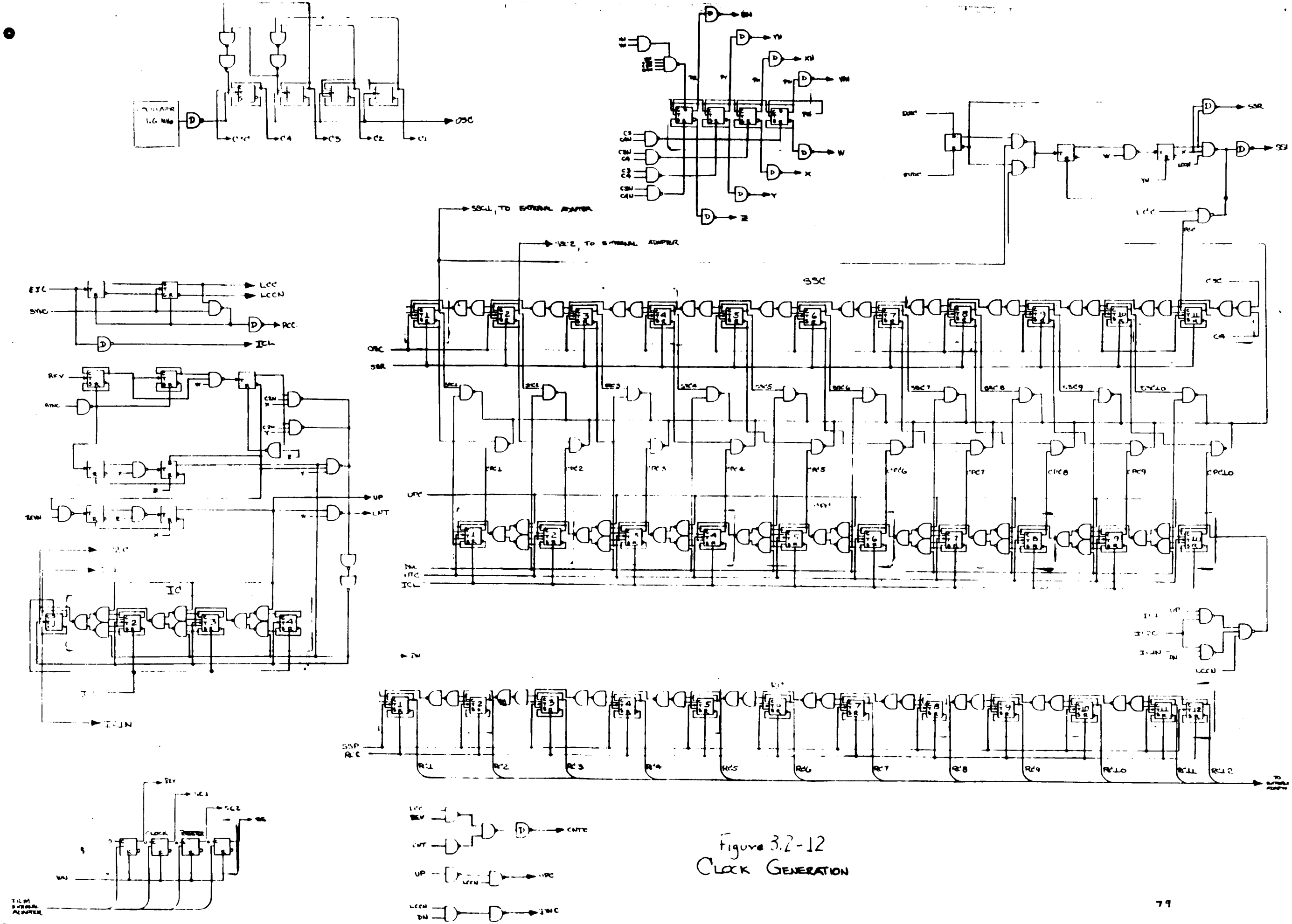


Figure 3.2-12
CLOCK GENERATION

Glossary

The Block Diagram (Figure 3.2-2) and the following Detailed Logic diagrams contain a number of data and control signals which are identified as mnemonics. Virtually all of these mnemonics have meanings which contribute to an understanding of the detail design. Table 3.2-4 lists a number of commonly used mnemonic suffixes which help to understand at a glance what kind of signal is being used. Table 3.2-5 lists most of the mnemonics used and their definition or derivation.

Table 3.2-4

Commonly Used Suffixes of Logic Mnemonics

C	Clear
C	Count
D	Decrement
E	Enable
G	Gate
H	High
I	Increment
L	Low
N	Not (Inverse side of flip-flop)
R	Register
S	Set

Table 3.2-5

Glossary of Logic Mnemonics

BMC	Begin Memory Cycle
BCM	Branch Mode Code
BR	Branch
BRC (n)	Branch Condition (n)
C1	Clock 1 - 800 KHz
C2	Clock 2 - 400 KHz
C3	Clock 3 - 200 KHz
C4	Clock 4 - 100 KHz
CAD	Channel Address Decrement
CNT	Count
CNTC	Count Compare
CPC	Compare Counter
DC	Data Clock
DIR	Discrete Input Register
DIRC (n)	DIR Clear (n)
DIRS (n)	DIR Set (n)
DN	Down
DNC	Down Compare

Table 3.2-5 (continued)

DOR	Discrete Output Register
DR	Data Ready
DRM	Data Ready Memory
E	Execute
EA	External Adapter
EE	End Execution
EI	External Initialize
EIC	External Initialize Clodk
EUDC	External Use Data Synchronous Clock
EUSC	External Use Space Craft Clock
FC	Fetch Command
FCS	FC Set
HI-X	High X Memory Address
HI-Y	High Y Memory Address
IAR	Input Address Register
IARC	IAR Clear
IARI	IAR Initialize
IM	Input Memory
IC	Integrating Counter
ICL	Initialize Clock
IF	Instruction Fetch
II	Initialize
INC	Increment

Table 3.2-5 (continued)

IR	Instruction Register
IRGL	IR Gate Low
IRGH	IR Gate High
IRAE	IR A Enable
IRBC	IR B Clear
IRBG	IR B Gate
IRCHC	IR C High Clear
IRHG	IR C High Gate
IRCLE	IR C Low Enable
IRDC	IR D Clear
IRDG	IR D Gate
IXM	Index Memory
IXR	Index Register
IXGL	Index Low
IXGH	Index High
IXRL	IXR Low
IXRH	IXR High
IXD	IXR Decrement
LCC	Load Compare Counter
LO-X	Low X Memory Address
LO-Y	Low Y Memory Address
MAR	Memory Address Register
MARC	MAR Clear

Table 3.2-5 (continued)

MARS	MAR Set
MARH	MAR High
MBR	Memory Buffer Register
MCR	Mode Code Register
MXB	Maximum Buffer Register
OAR	Output Address Register
OARS	OAR Set
OBC	Output Bit Clock
OC	Output Clock
OM	Output Memory
OMI	Output Memory Inhibit
OSC	Oscillator (1.6 MHz)
PAR	Program Address Register
PARS	PAR Set
PARC	PAR Clear
PC	PreCounter
PM	Program Memory
RBR	Read Buffer Register
RC	Revolution Counter
RCC	RC Clear
REV	Revolution System Clock
RI	Read Inhibit
RT	Route
RTE	Route Execute

Table 3.2-5 (continued)

SC1	System Clock 1
SC2	System Clock 2
SD	Set Discretes
SIX	Set Index Register
SOB	Set or Branch on Index
SS	Sub Sector System Clock
SSC	Sub Sector Counter
SSP	Sub Sector Pulse
SSR	Sub Sector Reset
SYNC	External Sync
T	Tape Control
TD	Times Decrement
TX	Transmitter
UP	Up
WC (n)	Wait Condition (n)
W, X, Y, Z	Phase clocks.

3.3 Memory

3.3.1 General

The memory for the S^3 Data Processing System is a modular, ferrite core, coincident-current storage system with a capacity of 3968 four-bit bytes, reducible to 1984 four-bit bytes by replacing the pluggable core array and diode matrix assembly and deleting several integrated circuit flatpacks and discrete components from the multilayer circuit board. The 2K and 4K byte memories use the same circuit board wiring.

The memory contains its own independent timing circuits. A Begin Memory Cycle pulse from the DPS logic external to the memory starts the memory read-store cycle which takes 3.2 microseconds. The short memory cycle allows maximum time for data manipulation in the DPS and is more than adequate in speed to store data at the maximum anticipated input data rates. Wherever possible, the memory uses standard DTL integrated circuits which have been proven for aerospace programs. In cases where standard DTL flatpacks are unusable because of voltage, power, or accuracy limitations, hybrid dual-transistor flatpacks and discrete components which have been qualified for aerospace environments are used.

The cores used for storage elements are a type developed by IBM for its 4 Pi family of aerospace computer memories. They are designed for a much wider temperature range than that anticipated for the S^3 program, and require only a slight amount of temperature compensation of drive currents for the S^3 application.

3.3.2 Design and Characteristics

The memory is organized as a coincident-current system. A single core plane containing 7936 cores in the 2K byte system and 15,872 cores in the 4K byte system, is wired in quadrants such that a combination of X and Y addresses selects four cores, one in each quadrant. Each quadrant contains a sense line running parallel to the Y-address lines, and an inhibit line running parallel to the X-address lines.

Therefore, the core plane is electrically similar to a stack of four small planes, each plane containing 1984 cores in the case of the 2K byte memory, and 3968 cores in the case of the 4K byte memory. In both sizes of memories, the number of X-coordinate wires remains constant at 62 per quadrant.

In the smaller memory, each quadrant contains 32 Y-coordinate wires, while in the larger memory each quadrant contains 64 Y-coordinate wires. Figures 3.3-1 and 3.3-2 are block diagrams of the 2K byte and 4K byte memories, respectively. Figure 3.3-3 shows the core and wire arrangement in the memory plane. Actually a few of the cores are deleted for the permanent load program where zeroes are required.

The memory cycle consists of a complete read-store cycle each time the memory receives a Begin Memory Cycle (BMC) pulse. A timing diagram is shown in Figure 3.3-4. The cycle is initiated by the falling edge of the BMC pulse. Prior to the beginning of the cycle, a pair of X-coordinate and a pair of Y-coordinate drivers are selected for coincident current at the location to be read and/or stored. Each coordinate driver has its own address input. The smaller memory requires 28 address lines, while the larger memory required 32 address lines.

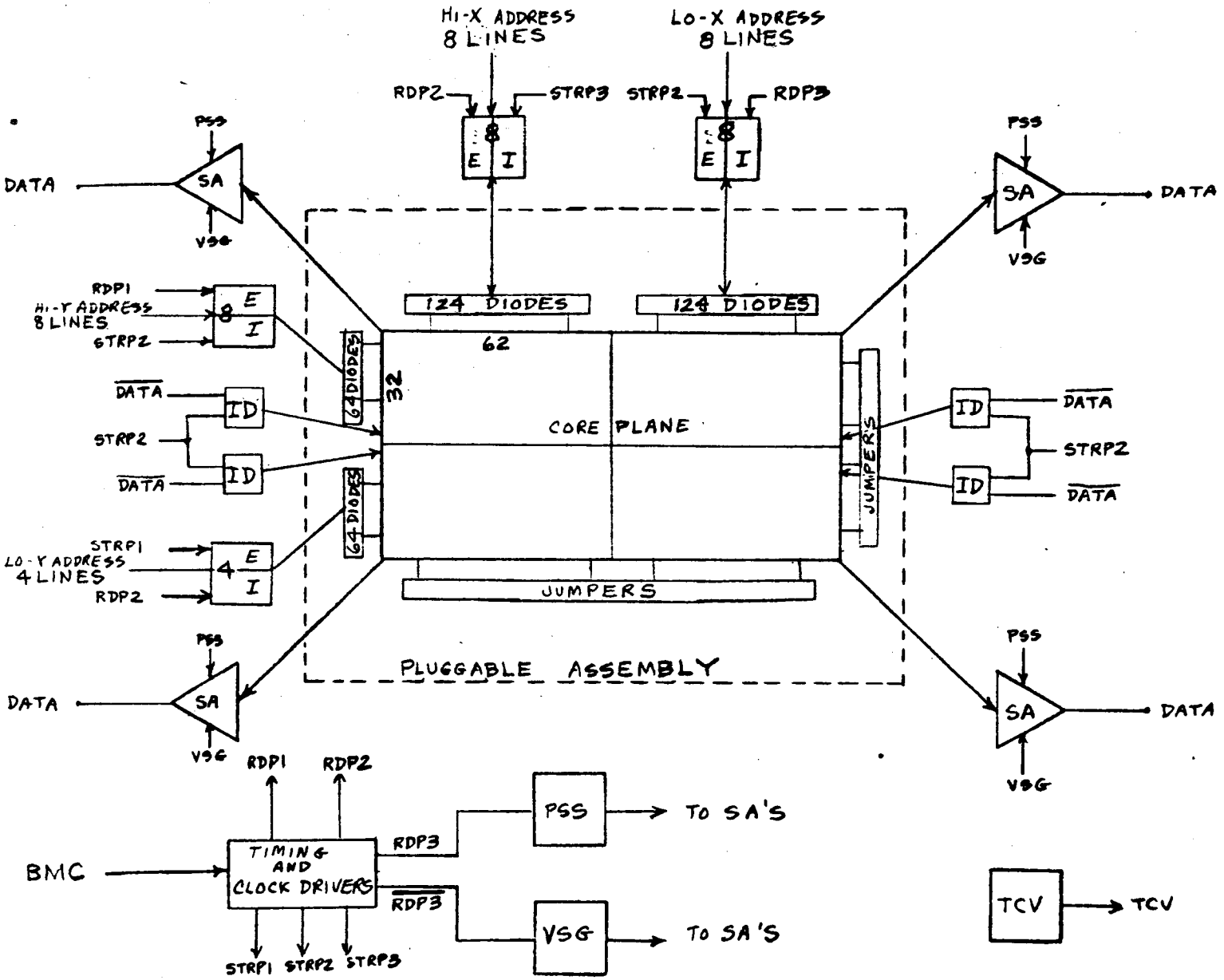


FIGURE 3.3-1 2K BYTE BY 4-BIT MEMORY

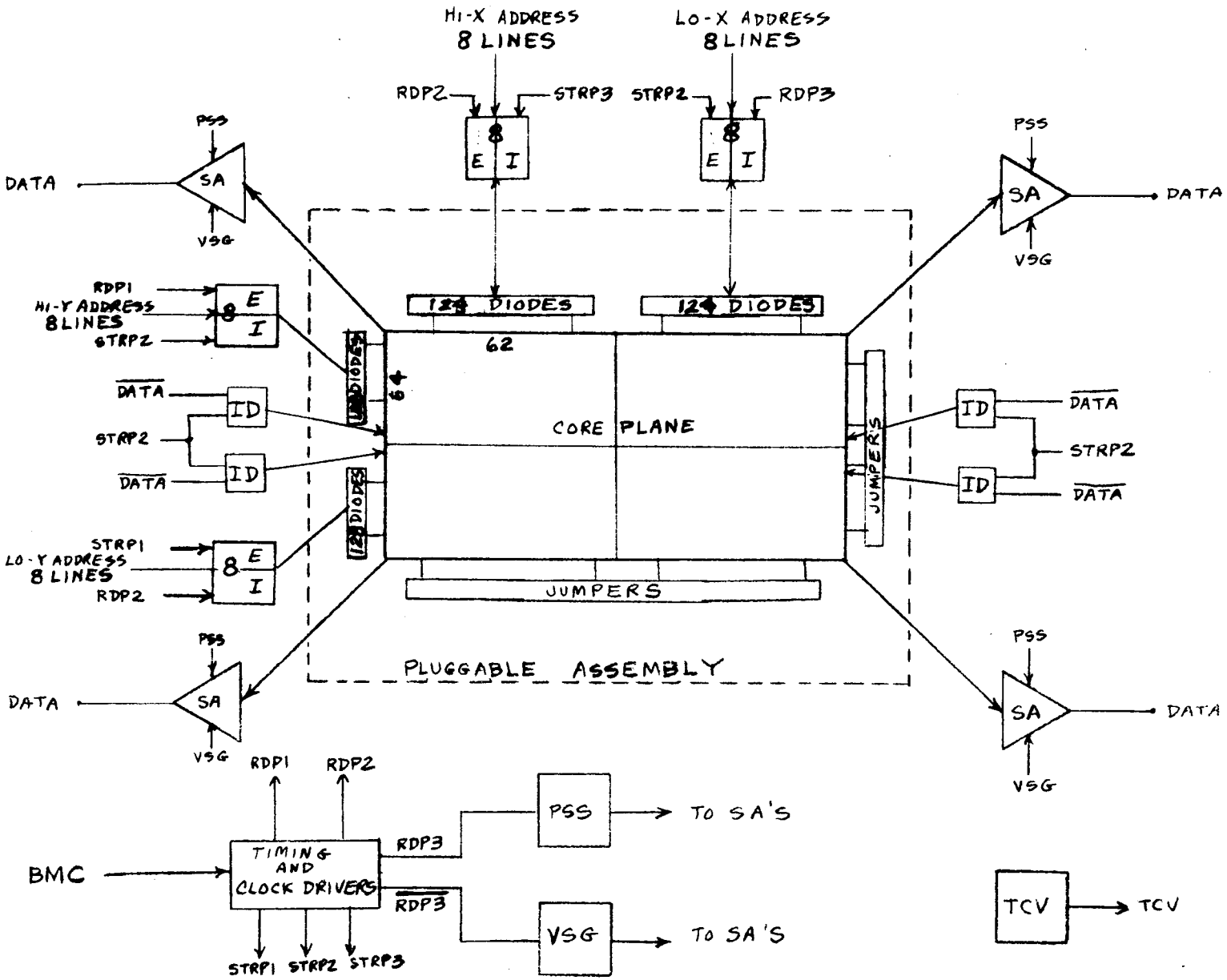


FIGURE 3.3-2 4K BYTE BY 4-BIT MEMORY

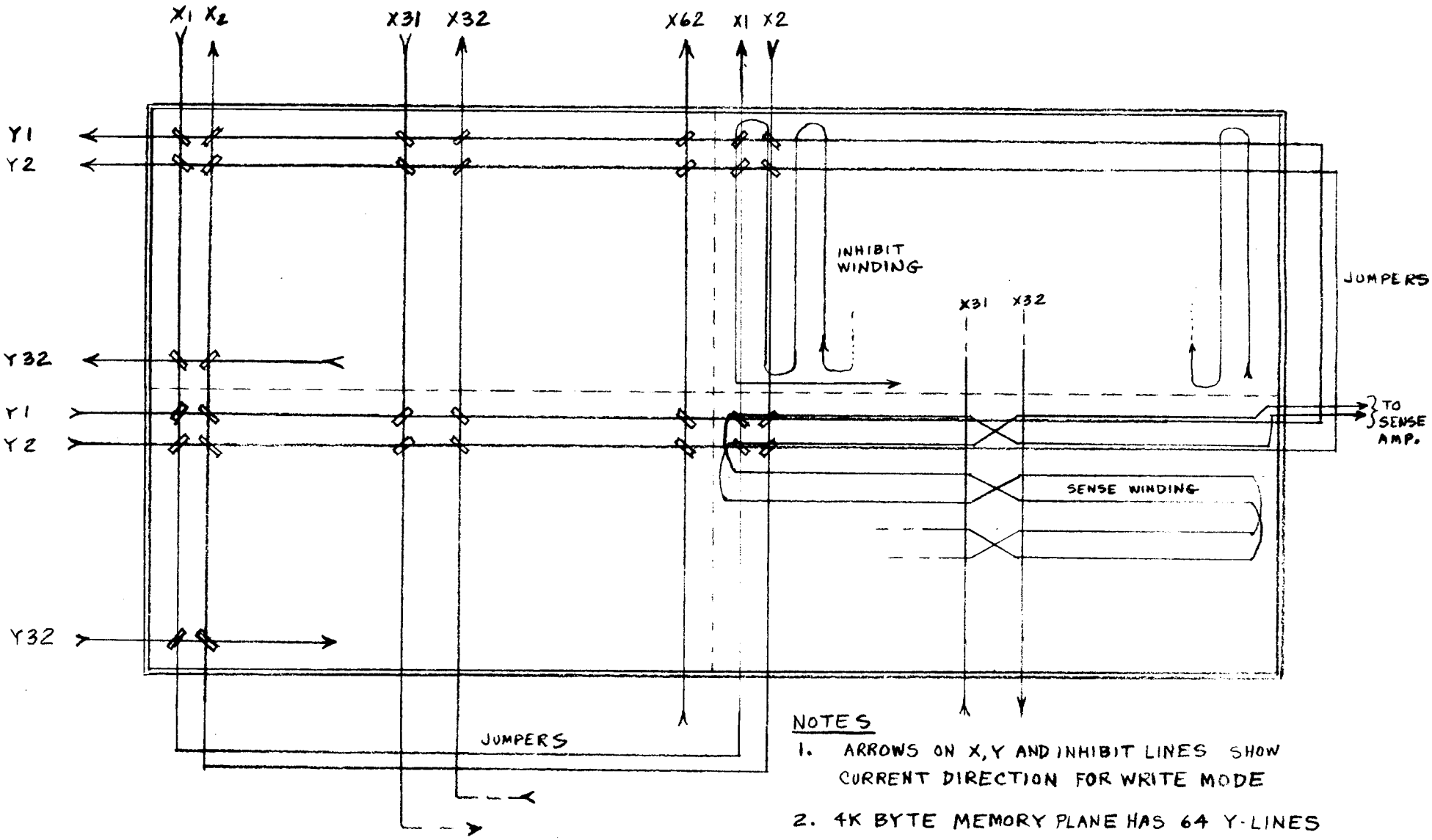


FIGURE 3.3-3 2K BYTE MEMORY PLANE SHOWING WIRE AND CORE ARRANGEMENT

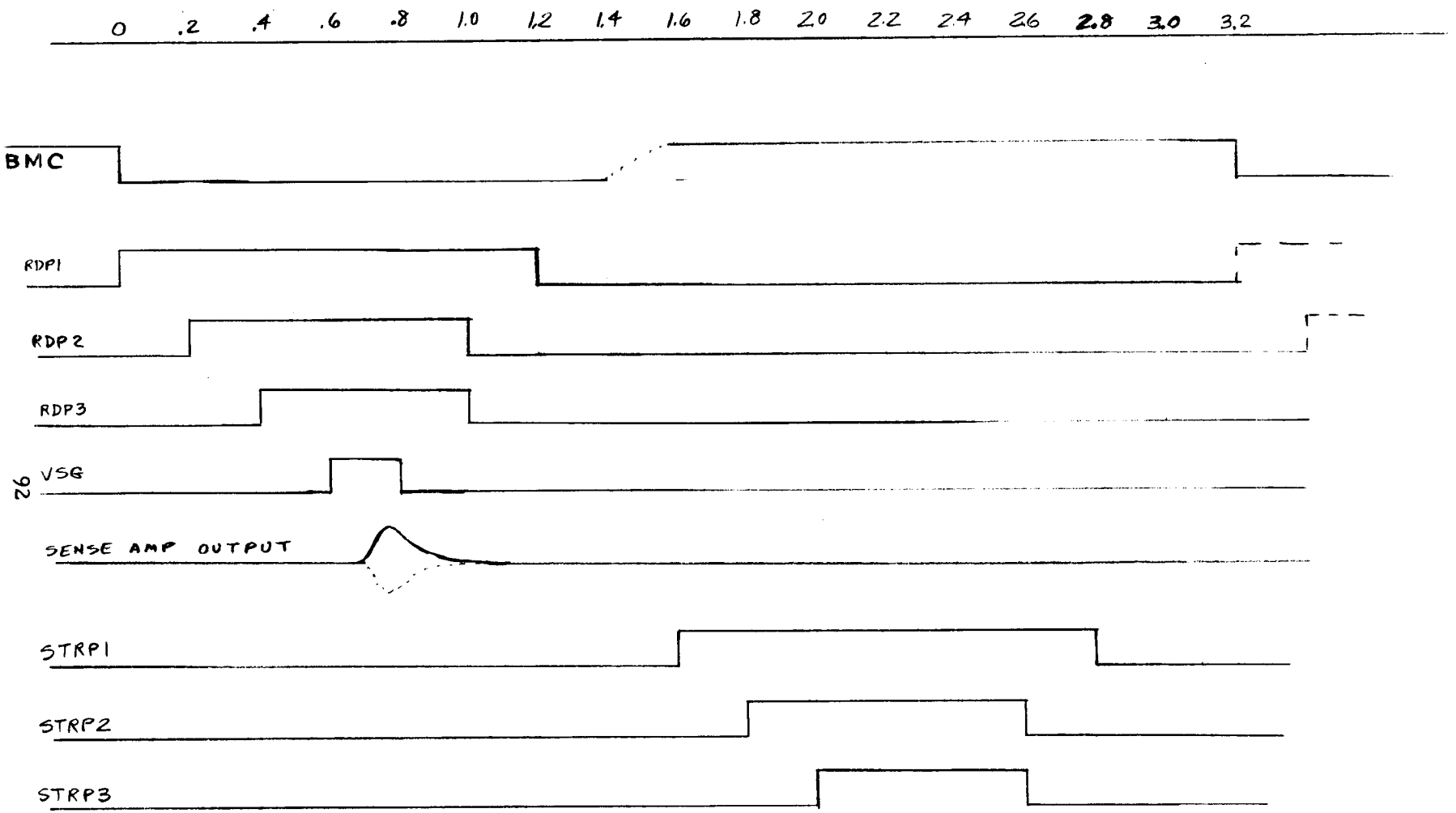


FIGURE 3.3-4 MEMORY CYCLE

At the fall of the BMC pulse, the timing circuits and clock drivers generate three overlapping read pulses (RDP1, RDP2, and RDP3), which turn on the selected coordinate drivers, driving half-select currents through the selected X and Y lines, applying full-select currents to only one core in each quadrant of the plane. During the RDP3 pulse, operating power is applied to the sense amplifiers. Shortly after the start of the RDP3 pulse, a variable strobe gate (VSG) strobes the sense amplifiers, which read out the four data bits in parallel into logic external to the memory.

During the latter half of the memory cycle, three overlapping store pulses (STRP1, STRP2, and STRP3) are generated which cause half-select write currents to flow in the selected X and Y-lines. The inhibit drivers control whether a "one" or a "zero" is to be stored in each location. If a "zero" is to be stored, the inhibit driver is turned on, generating a current which cancels out the X half-select current. Since the net current through the core is only the Y half-select current, the core remains in a "zero" state.

The inhibit drivers are controlled by the external Central Control logic, which determines whether the data read out shall be restored or new data stored.

Because the switching threshold of the cores varies with temperature, the X, Y, and inhibit currents are compensated by a temperature controlled voltage (TCV) circuit.

Figure 3.3-5 is a graph showing estimated average memory operating power versus address time. This graph is the result of an initial tradeoff study of 2K and 4K byte memories with 3 bits and 4 bits per byte, and is based on initial estimates of memory circuit requirements. Since that study, a closer look at each circuit has resulted in

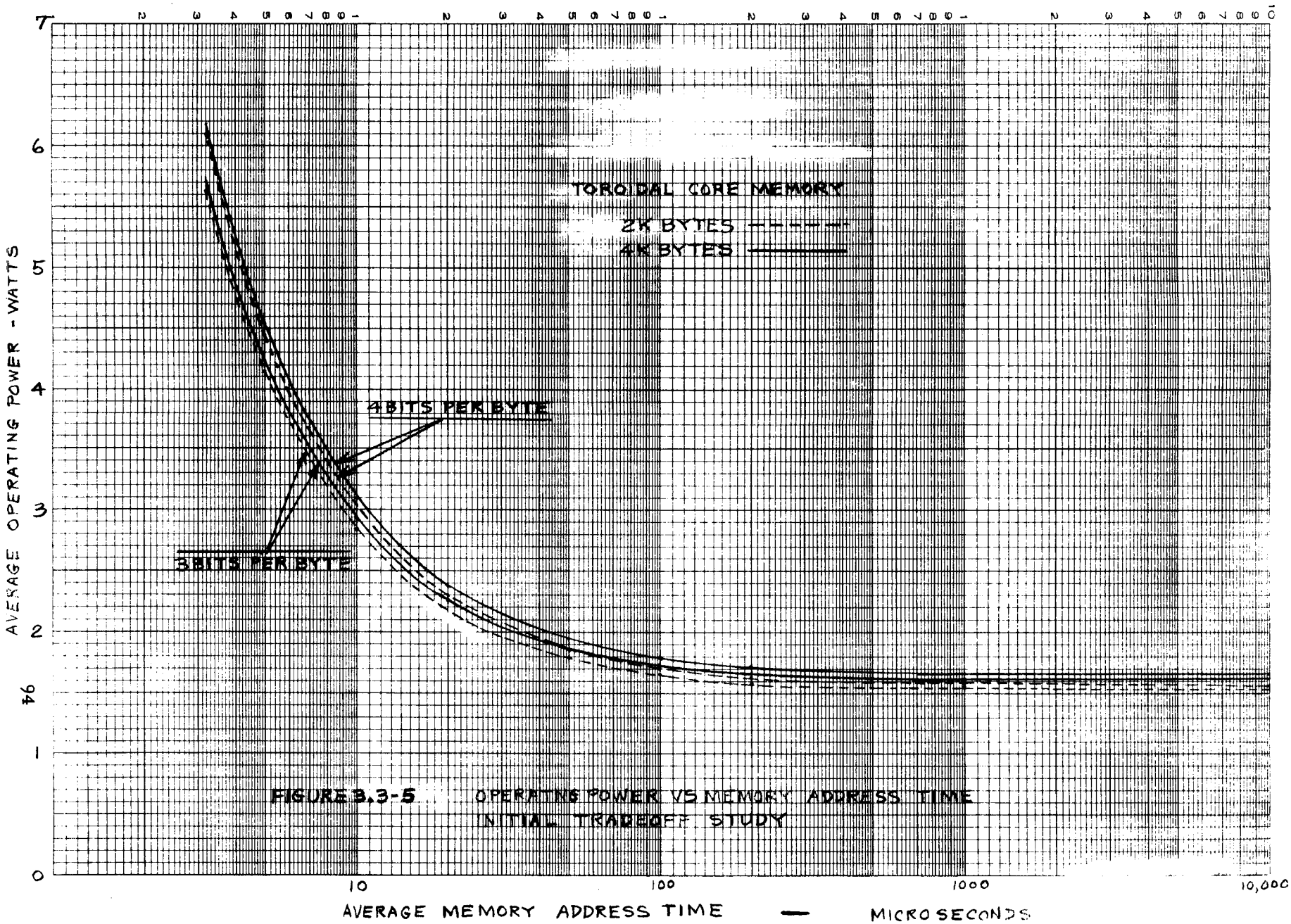


FIGURE 3.3-5 OPERATING POWER VS MEMORY ADDRESS TIME
INITIAL TRADEOFF STUDY

the addition of several logic blocks, with a consequent slight increase in power of 80 to 90 milliwatts on both the 2K and 4K byte memories. However, Figure 3.3-5 is adequate for comparative purposes. It should be noted that there is negligible difference in power between a 3 bit-per-byte and a 4 bit-per-byte memory even though, with the same number of bytes, the 4 bit-per-byte memory has 33% more bits of storage.

The original calculated standby powers were as follows:

2K X 3 Bits: 1.52 watts

2K X 4 Bits: 1.57 "

4K X 3 Bits: 1.60 "

4K X 4 Bits: 1.65 "

Figure 3.3-6 shows the memory power based on the revised circuit designs for the 2K and 4K byte, 4 bit memories. Standby powers are as follows:

2K X 4 Bits: 1.66 watts

4K X 4 Bits: 1.73 watts

The operating energy per read-write cycle is based on the timing diagram of Figure 3.3-4 and is 14.5 microwatt-seconds for either size of memory. To obtain the operating power curves of Figure 3.3-6, the operating energy was divided by the memory address time and added to the standby power. At the minimum expected operating temperature of -10°C, the operating energy is approximately 8% higher. However, at the average memory address period in S^3 applications of not less than 100 to 1000 microseconds, the operating power is very nearly standby power, and therefore the average operating power may be considered constant over the entire memory temperature range. Analysis of the S^3A application (Section 3.10) revealed an average memory address period of 1,188 μsec for a worst case program implementation and including program and data input and output operations.

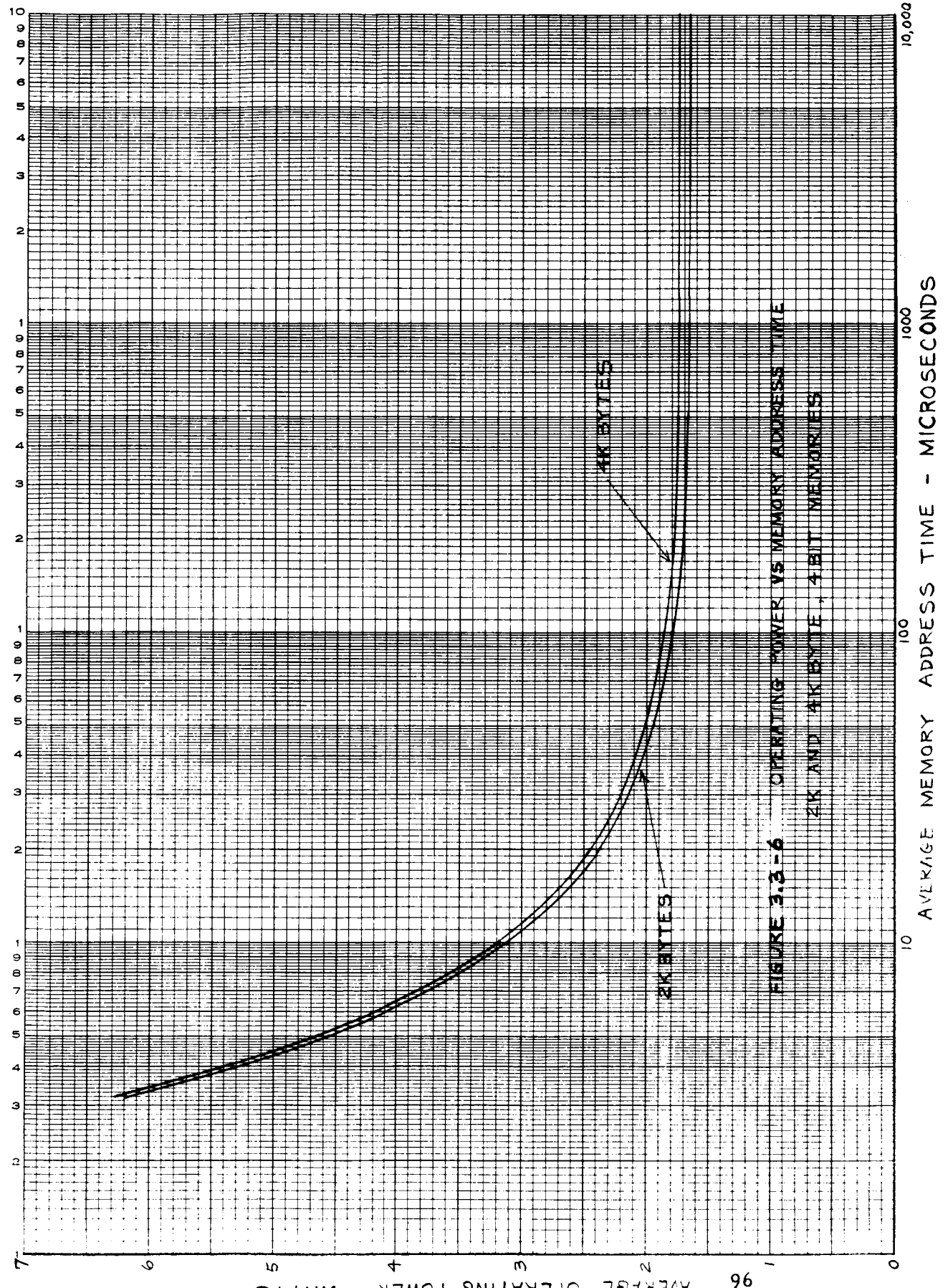


FIGURE 3.3-6 OPERATING POWER VS MEMORY ADDRESS TIME

All figures are based on nominal power supply voltages.

Power Supply Requirements

Power supply voltages required are +5vdc, +12vdc, and -6vdc. Tolerances are ±5% on all supplies.

2K byte memory module current requirements are:

+5v @ 300 ma

+12v @ 6 ma (1000 usec address time)

-6v @ 16 ma

4K byte memory module current requirements are:

+5v @ 315 ma

+12v @ 6 ma (1000 usec address time)

-6v @ 16 ma

The +5v and -6v current requirements are constant. The +12v average current varies with memory duty cycle. A graph of average +12v supply current versus address time is shown in Figure 3.3-7.

Peak +12v supply currents are shown in Figure 3.3-8. The external power supplies should have sufficient filtering to be capable of supplying these peak currents with ±5% regulation.

Input/Output Data

The total number of I/O lines at the memory delta pack connector is itemized below.

For the 2K byte module:

28 Address

1 BMC pulse

4 Data in

4 Data out

3 Power Supplies

2 Ground Returns

42 separate lines total.

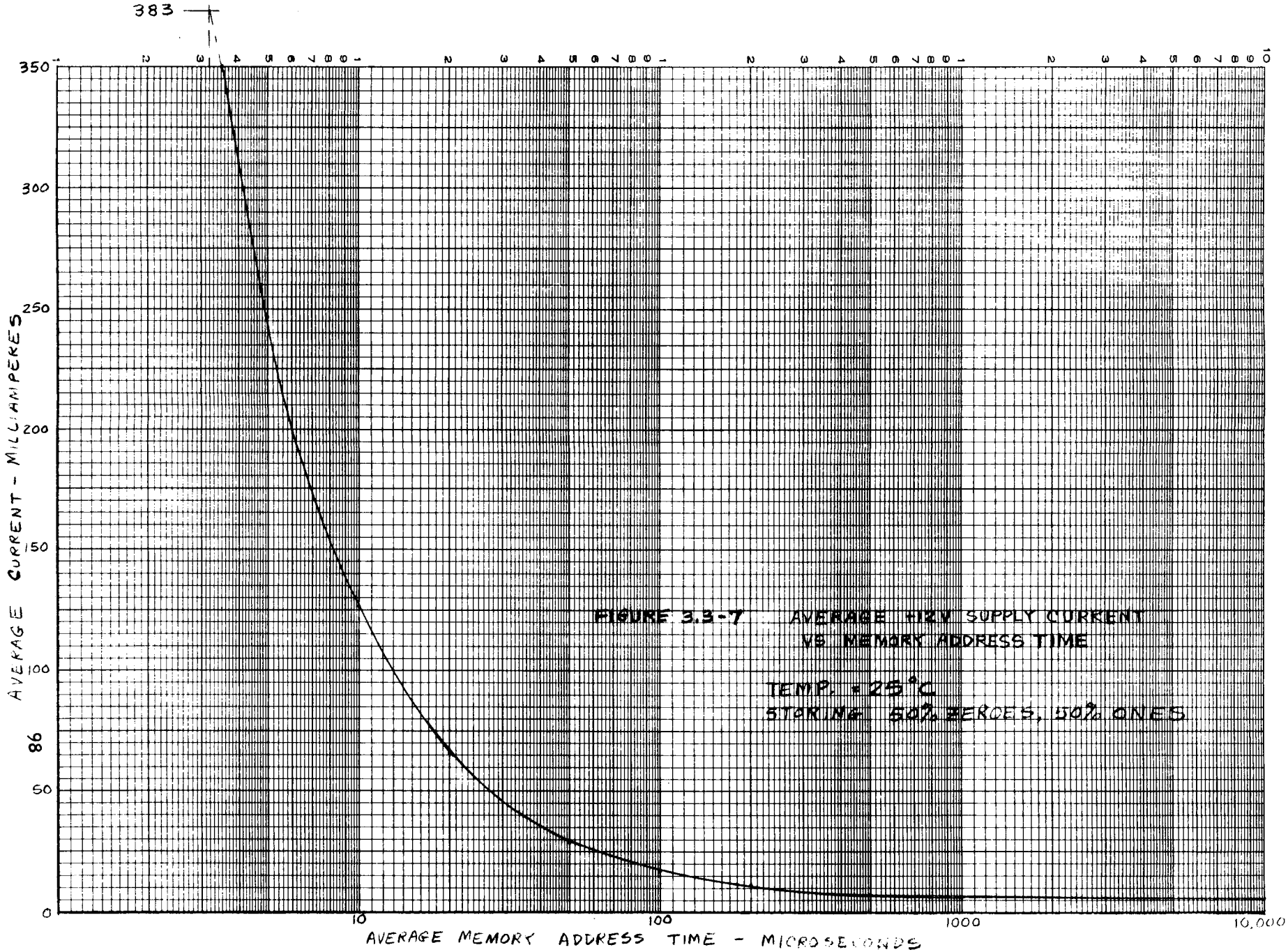


FIGURE 3.3-7 AVERAGE +12V SUPPLY CURRENT VS MEMORY ADDRESS TIME

TEMP. = 25°C
STORING 50% ZEROES, 50% ONES

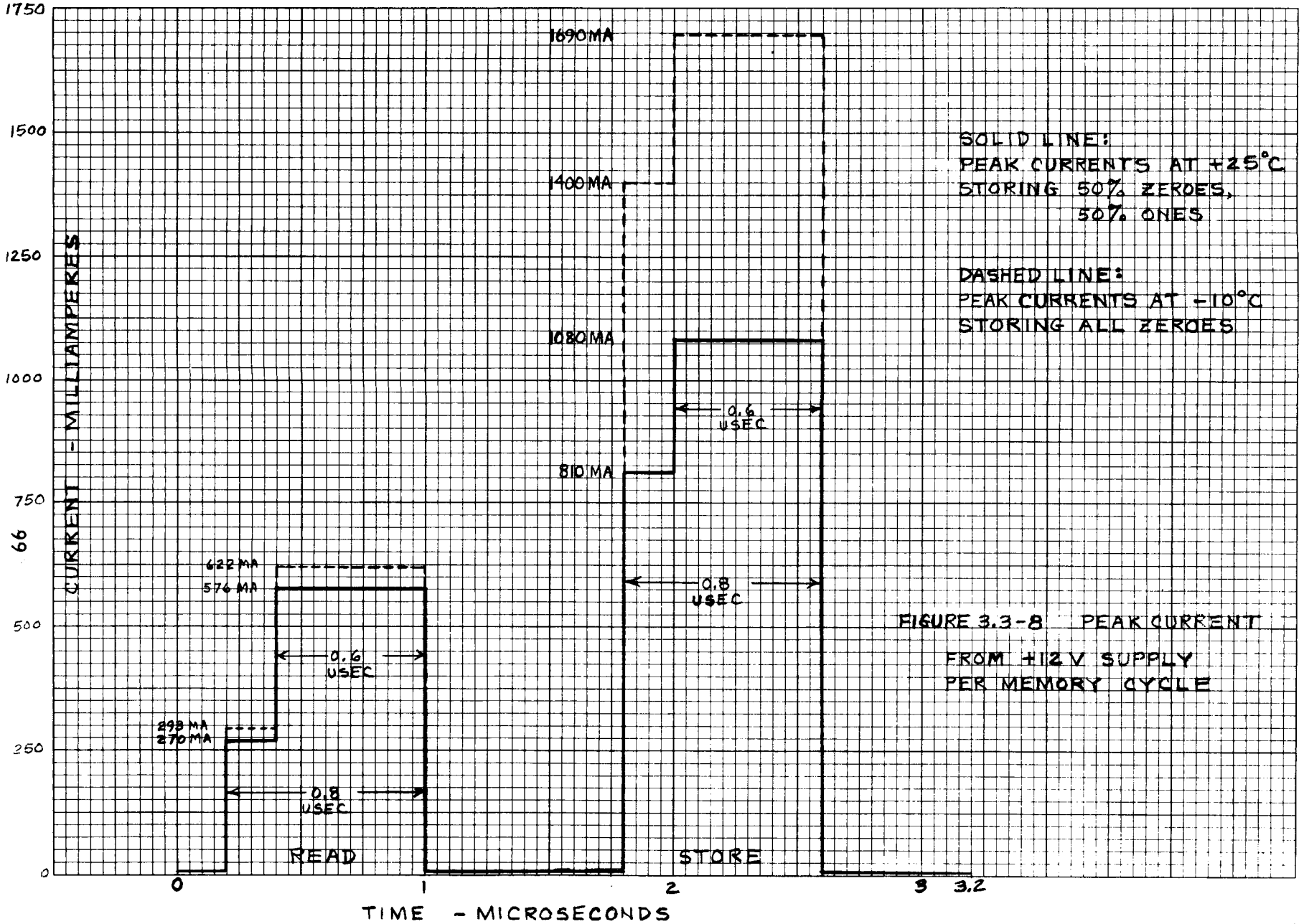


FIGURE 3.3-8 PEAK CURRENT FROM +12 V SUPPLY PER MEMORY CYCLE

For the 4K Byte module:

32 Address

1 BMC pulse

4 Data in

4 Data out

3 Power Supplies

2 Ground Returns

46 Separate lines total.

Several connector pins will be paralleled for the power supply and ground connections. The ground return for the high current circuits (the coordinate driver and inhibit driver output circuits) will be kept separate from the other ground returns.

Since power supply requirements have already been discussed, the following section describes the input and output signals.

Input/Output Signals (All 0v or +5v logic levels)

Inputs

(1) Address: 28 lines for 2K byte memory.
32 lines for 4K byte memory.

Levels: A down logic level selects a coordinate driver

Loads: Each line is one LPDTμL Load.

Timing: Four coordinate drivers must be selected before BMC signal is initiated.

(2) Begin Memory Cycle: One line for either memory size.

Level: Down logic level (falling edge) starts memory cycle. Pulse width is not critical.

Load: 4 DTμL Loads

Timing; Fall time must be 25 nanoseconds per volt or faster. Rise time is not critical.

(3) Data Inputs: 4 lines for either memory size.

Levels: An up logic level selects an inhibit driver for storing a zero. Therefore the complement of the data word is required as an input.

Loads: Each line is one DTμL load.

Timing: Data bits must be present no later than 1.8 microseconds after start of memory cycle. Pulse width: Sufficient to overlap the memory STRP2 pulse.

Outputs: 4 lines for either memory size.

Levels: An up logic level denotes a "one".

Loads: Each line will drive 4 LPDTμL loads maximum

Timing: Data pulses will occur 0.7 to 1.0 microsecond after start of cycle. Pulse width: to be determined for compatibility with LPDTμL data register.

Memory Circuits

The memory circuits employ monolithic and hybrid integrated circuits to the highest degree practicable. The monolithic circuits are DTL and low power DTL (LPDTμL) devices. Hybrid circuits include dual NPN and PNP transistor flatpacks and multiple-diode flatpacks for the core selection matrices. Discrete resistors, zener diodes, capacitors, and other components are used where voltage, current, and timing accuracy is required. Detailed descriptions of each circuit follow.

Timing and Clock Drivers

The timing and clock driver circuit is shown in Figure 3.3-9. Integrated flatpack monostable multivibrators (DTμL 951) are used to generate the read and store pulses and delays shown in Figure 3.3-4. The read and store pulses drive the coordinate drivers, inhibit drivers, variable strobe gate, and power supply switch. For fast rise and fall times

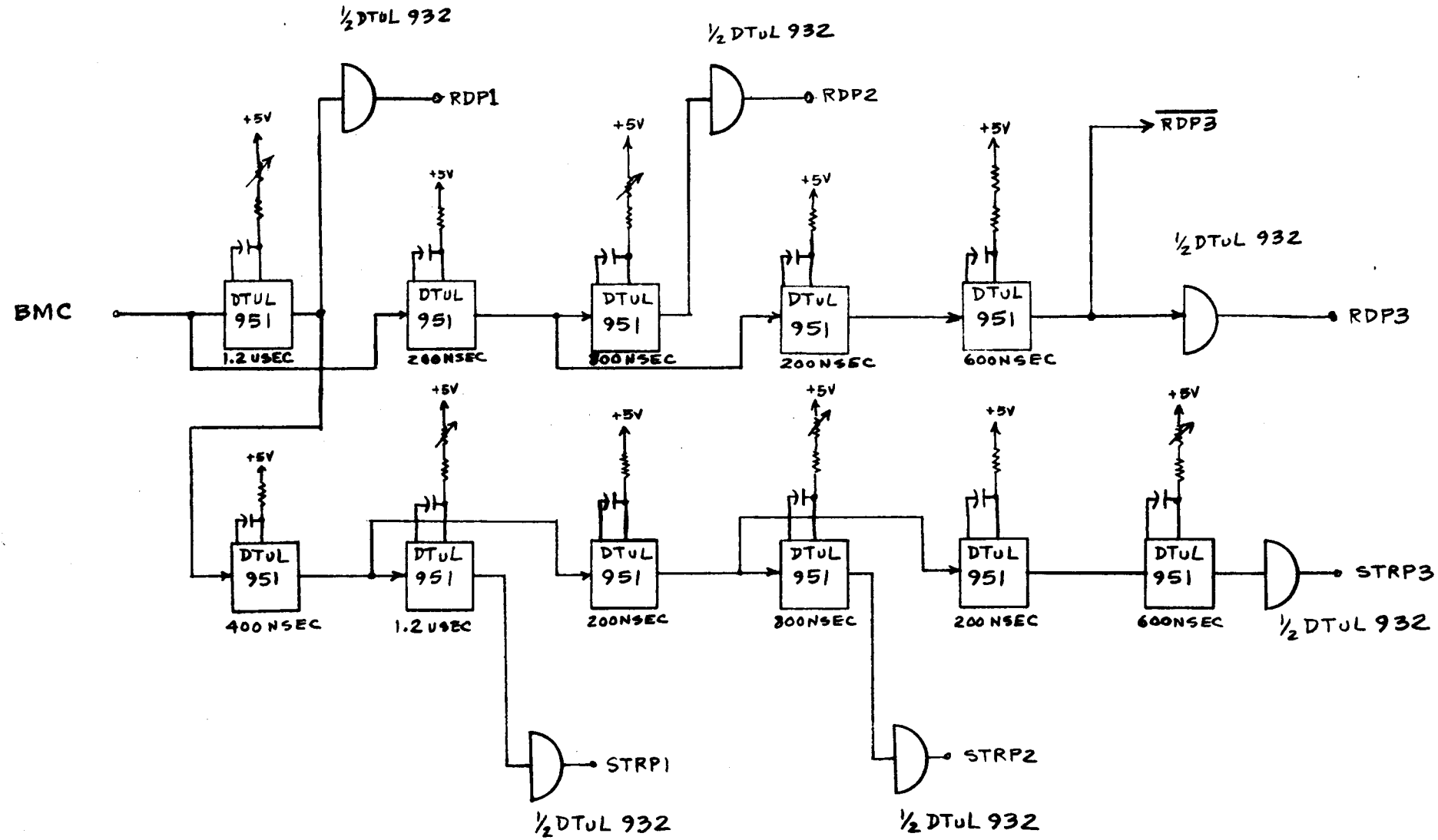


FIGURE 3.3-9 TIMING AND CLOCK DRIVERS

when driving multiple loads with shunt capacity, the outputs are buffered by DTμL 932 power gates, which have an emitter follower pull-up transistor. The pulse widths are adjustable by means of the potentiometers shown, while the delays, which are not critical, are fixed.

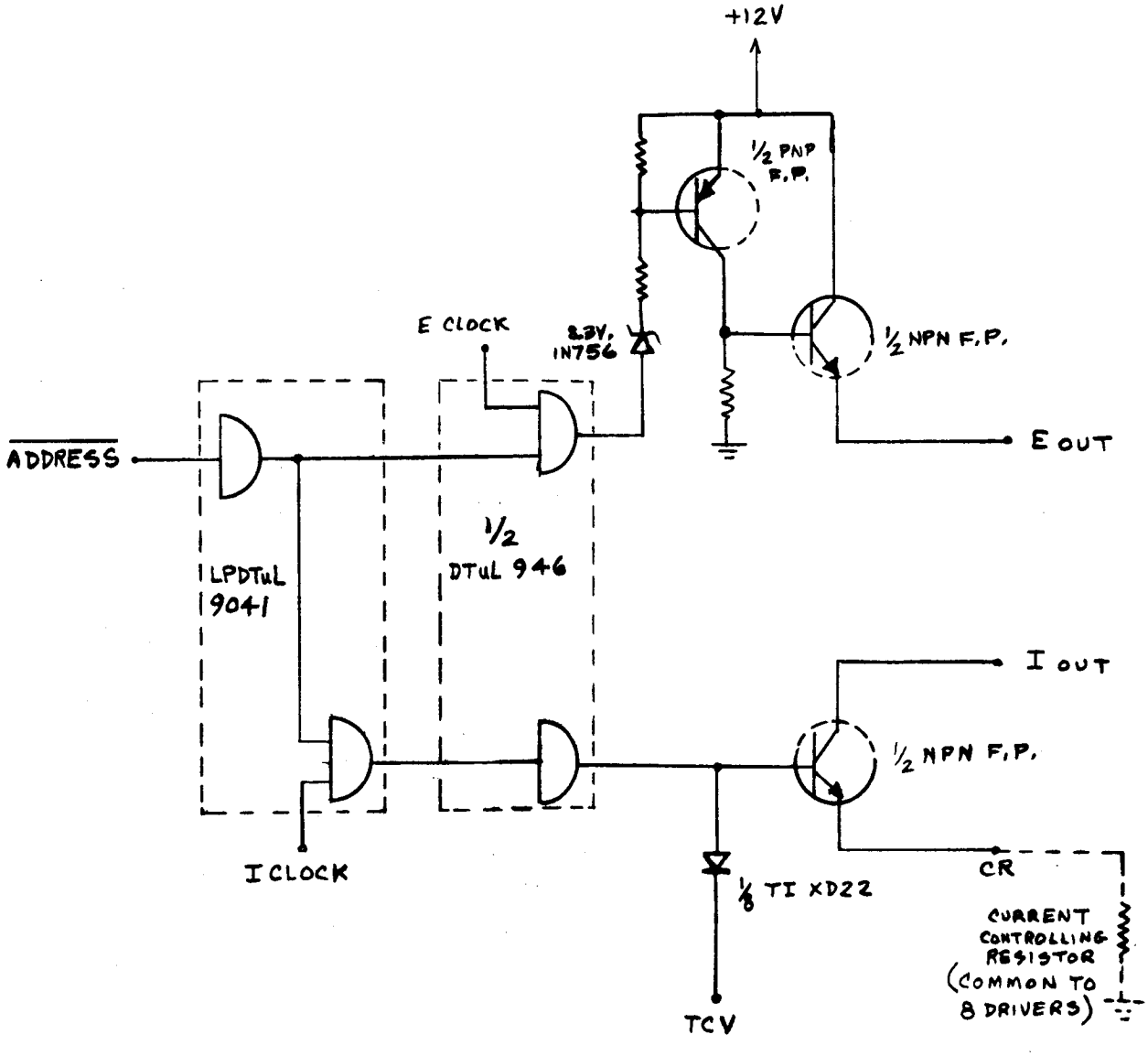
The complete memory read-store cycle is initiated by the falling edge of the BMC pulse.

Coordinate Driver

The coordinate (EI) drivers, Figure 3.3-10, drive the X and Y coordinate wires of the selected core location. Each EI driver consists of a voltage source (E) and a current sink (I). A down level at the address input selects the EI driver. When a particular EI driver is selected, no action occurs until an E clock or an I clock pulse is received. The E clock and I clock pulses are the "read" and "store" pulses from the timing and clock driver circuits. The memory block diagrams, Figures 3.3-1 and 3.3-2, indicate which read and store pulses drive which EI drivers.

When an E clock pulse is applied to a selected EI driver, the "E" output transistors conduct, applying the positive supply voltage to a group of X or Y wires through a selection diode matrix. When an I clock pulse is applied to a selected EI driver, the "I" output transistor conducts, drawing current from one X or Y wire through a selection diode matrix. The voltage to which the base of the output transistor rises is controlled by the TCV circuit. A precision resistor in the emitter of the output transistor determines the emitter current and therefore the collector current.

A combination of four EI drivers, two acting as voltage sources and two as current sinks, is required to address a particular location. Since only one EI driver in each coordinate group is operated at any one time, only one emitter resistor is required for each coordinate group.



NPN TRANSISTORS SIMILAR TO 2N2222
PNP TRANSISTOR SIMILAR TO 2N2412

FIGURE 3.3-10 COORDINATE DRIVER (EI)

Inhibit Driver

The inhibit driver (ID), is similar to the I portion of the EI driver circuit and is shown in Figure 3.3-11. When a zero is to be stored, the inhibit driver must be turned on. This is caused by inserting a "1" and the STRP2 pulse at the inputs to the logic gate. Therefore, the inhibit driver requires the complement of the data word. Since it is possible that all inhibit drivers will be conducting simultaneously, each circuit requires its own emitter resistor which determines the inhibit current.

Variable Strobe Gate

The variable strobe gate (VSG) is shown in Figure 3.3-12. This circuit generates a positive pulse which strobes the sense amplifiers at the time when core responses are expected. An adjustable delay is initiated by the $\overline{\text{RDP3}}$ pulse. The adjustable delay is obtained with a DTμL 951 monostable multivibrator and external timing components. The fixed strobe width is generated by a second DTμL 951 unit with fixed external timing components. The pulse is buffered by a DTμL 932 power gate and drives the four sense amplifiers simultaneously.

Sense Amplifier

The sense amplifier (SA), Figure 3.3-13, is a Fairchild μA 711 dual comparator with external resistors which set the threshold sensitivity to distinguish between "one" and "zero" outputs from the memory.

A number of these units have been used by IBM as sense amplifiers in a prototype memory having characteristics similar to the S^3 memory. The input offset was measured individually on each unit and external resistors were selected to compensate for this offset. After installation the units retained their threshold sensitivity over the memory operating temperature range.

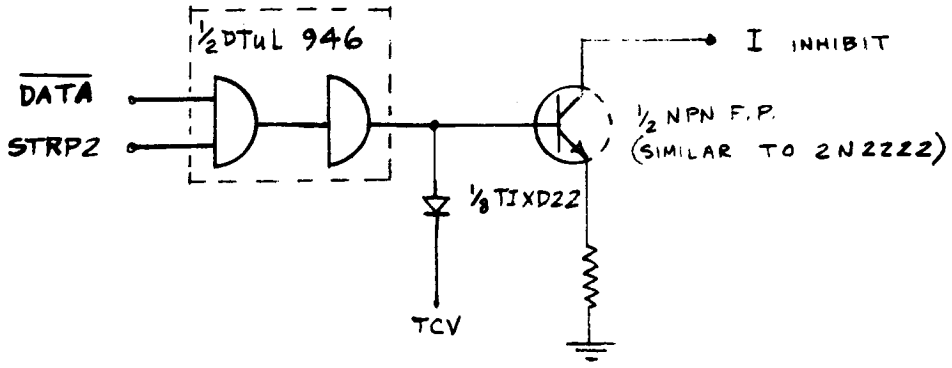


FIGURE 3.3-11 INHIBIT DRIVER (ID)

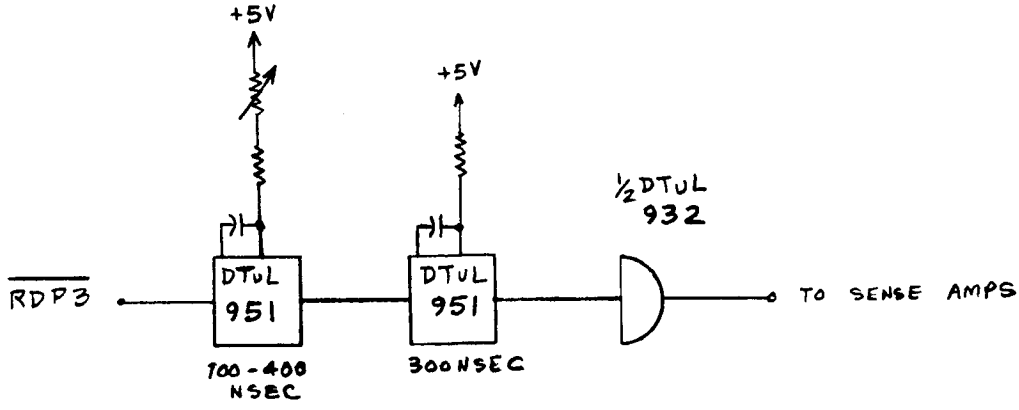


FIGURE 3.3-12 VARIABLE STROBE GATE (VSG)

The μA 711 possesses the advantage that the positive supply voltage can be reduced during standby, thus greatly reducing sense amplifier standby power.

Just before the sense amplifiers are to be strobed, full positive voltage can be applied without causing output transients from the amplifiers. Thus standby power is 40 milliwatts per amplifier compared to 130 milliwatts for a 0.6 microsecond period during readout when full power is applied.

The shunt capacitor on the output acts as a pulse stretcher to widen the output pulse sufficiently to operate the external low power DTL circuits.

Power Supply Switch

The power supply switch (PSS) is shown in Figure 3.3-14. This circuit is used to gate the positive power supply of the sense amplifiers on during readout. During the RDP3 pulse, the PNP transistor is turned on, applying the +12V supply to the positive supply input of the sense amplifiers. Whenever the RDP3 pulse is absent, the PNP transistor is open, and the sense amplifiers receive +5V on their positive terminal. Thus sense amplifier standby power is kept low, and the transition from operate to standby power and vice versa causes no transients on the sense amplifier output.

Temperature Controlled Voltage

The temperature controlled voltage (TCV) circuit is shown in Figure 3.3-15. This circuit clamps the positive base voltage levels of the coordinate driver and inhibit driver current sink transistors to a voltage which varies inversely with temperature. The base voltage control, in conjunction with the precision resistors in the emitter circuits of the current sink transistors, controls the half-select currents through the cores in order to track the core switching threshold which varies with

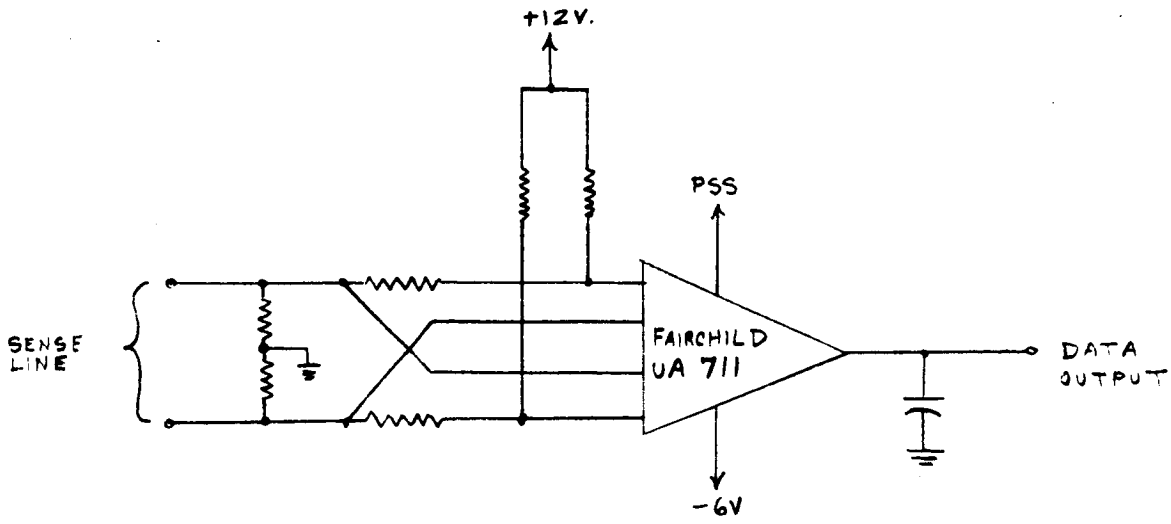


FIGURE 3.3-13 SENSE AMPLIFIER (SA)

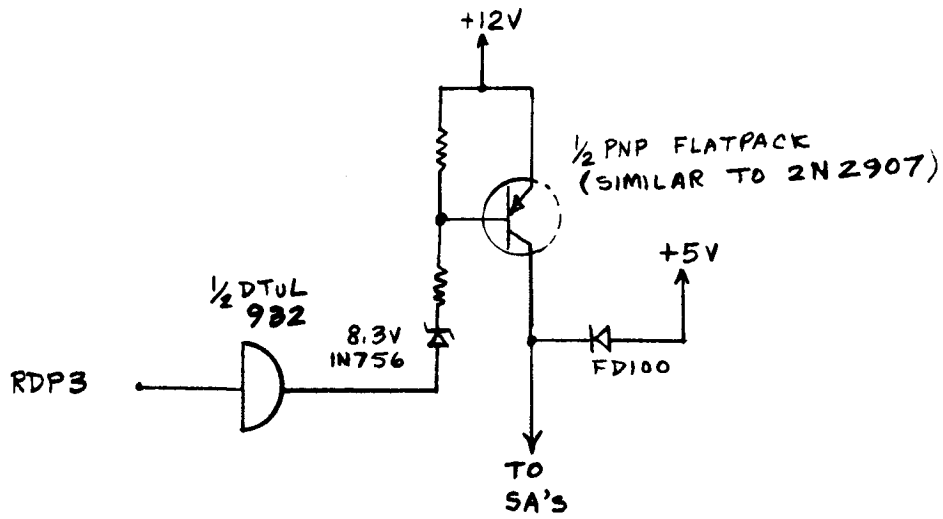


FIGURE 3.3-14 POWER SUPPLY SWITCH (PSS)

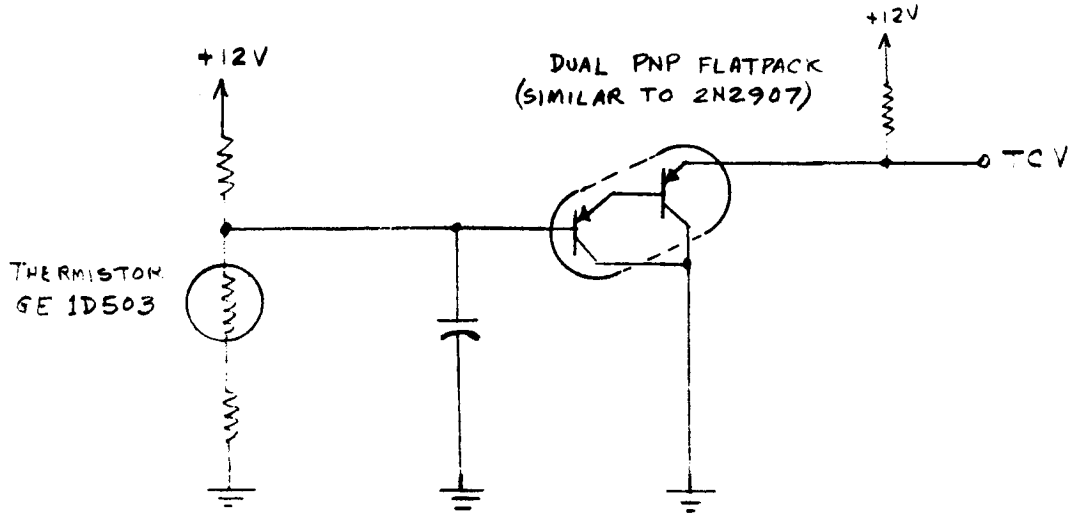


FIGURE 3.3-15 TEMPERATURE CONTROLLED VOLTAGE (TCV)

temperature. Since the core switching characteristics vary only slightly over the S^3 temperature range, a thermistor-controlled voltage is adequate for this purpose.

3.4 I/O Section

3.4.1 General Description

The I/O section interfaces the Central Control section with the satellite data sources. The I/O section performs, as instructed, the following functions:

- Data channel selection
- Random serial pulse accumulation (binary counting)
- Logarithmic data compression
- Parallel to serial conversion
- Analog to digital conversion
- Frame sync and identification synthesis
- Discrete function storage and output
- Serial data read out to central control

The data source types are:

- 0 to +5 volt analog
- Serial digital
- Parallel digital
- Random serial pulses, up to 1 MHz

A brief functional block diagram of the I/O section is shown in Figure 3.4-1.

The I/O accommodates up to 64 addressable data channels of analog, pulse, and serial digital signal types. These types can be accommodated in modules of eight each up to a total of 64 of any combination which does not require more than 32 analog channels. Six channels of serial digital data are reserved for message header data and associated hardware. The message header includes 24 bits of sync group, 8 bits of frame count for up to 256 different frames in a sequence,

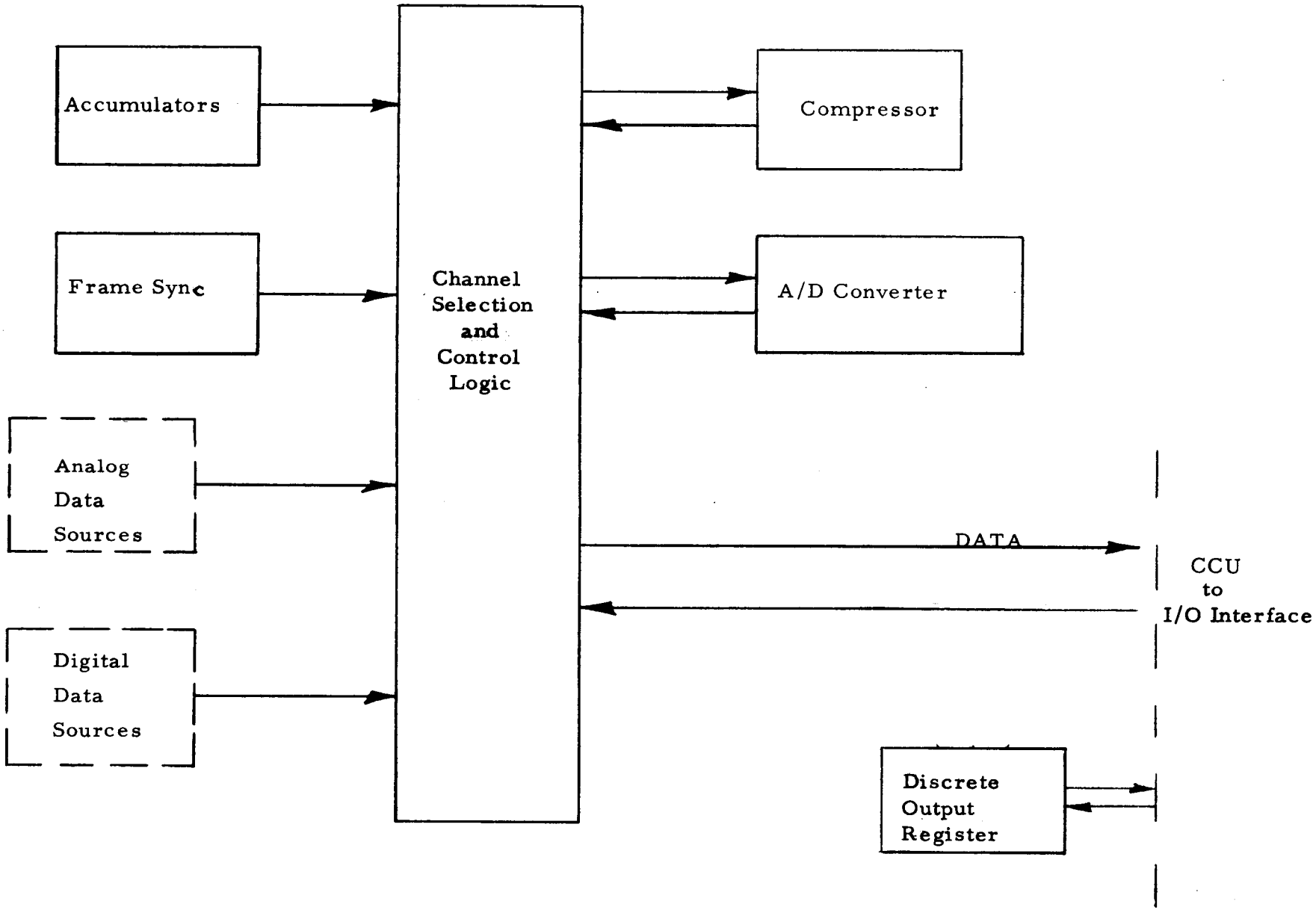


Figure 3.4-1 I/O Block Diagram

12 bits of relative time to the nearest 0.005 second, and a 4-bit identification tag. There are three different identification tag registers available which, together with frame count (which is program resettable) permit considerable latitude in the number of different formats usable at any one time; additional registers can be accommodated. In addition to addressable channels there are control and instruction provisions for accommodating subcommutating subchannels.

The I/O includes analog-to-digital conversion, accumulator/compressor hardware and provision for serial digital data. Accumulators are 20 bits long and the data is compressed under program control to floating point quantities of 8 or 9 bits to provide a 3% accuracy for 12 or 20 bit counts (counts of 4K or 1M). Six accumulators are provided. The A/D converter provides 8 or 10-bit digital quantities accurate to $\pm 1/2$ LSB under program control; it can also, under program control provide the 8 least significant bits (LSB's) of a 10-bit conversion for additional data compression capability to the 0.1% accuracy given that the full range is determined sufficiently often.

The I/O section includes 5 8-bit discrete output registers controlled by the program. These discretes can be operated, under program control, in a steady-state or pulsed mode. The discretes can be individually set or cleared and many can be set or cleared simultaneously. Any number of bits of a register may be set or cleared simultaneously. The same bit positions of any number of registers can be set or cleared simultaneously, and all 40 discretes can be set or cleared simultaneously with one command.

The discretes are assignable by the user and can be used for controlling experiment modes by resetting or triggering events, and for inputting back into the DPS via the branch inputs such as for program linkage.

The I/O design philosophy is that of providing a standard design of maximum capability and flexibility. The standard design is based on the maximum number of channels, the ability to process analog random pulses, and digital data in any of the specified ways, and an ability to allocate channels in modules of 8 to any type of data. For flight-to-flight variability in channel assignments, adaptive interfaces are provided to personalize the I/O section. In addition the discrete outputs are available for connection to the Central Control, external systems, or the I/O section itself.

The design adopted is one in which there are three levels of building blocks. The first level block represents the basic logic for recognizing and interpreting a request for data, and generating all of the necessary signals which all other blocks may use to fulfill the specific data request. The second level blocks are logic elements which perform specific functions, but which might not be required for all missions (compressor, a/d converter, differential amp., etc.). The third level blocks are those which are added or subtracted singularly as the number and type of data channels is varied (multiplexor channels, accumulators, parallel to serial conversion gates, analog switches, etc.). In some cases, the third level blocks are grouped for convenience, but in all cases the groups are deletable or expandable on an individual block basis.

It should be realized that, because one hundred forty-two DTμL packages can be assembled on a single MIB, a modular "plug in" block concept is impractical for this application. As demonstrated in Section 3.7, the entire CCU and I/O for a 64 channel DPS with six 20-bit accumulators, compressor, A/D converter, and 40 discrete outputs, can be packaged in three delta pacs. Figure 3.4-2 shows the fractional MIB area that each I/O block occupies and the power it consumes; note that these figures are for a maximum type of implementation.

Figure 3.4-2

I/O Size and Power Data

	DTμL Packages	MIB Board Sides	Power (MW)	Other
1. Discrete Storage (5 DOR's 40 Bits)	45	.32	180	
2. A/D	16	.31	500	1-810, Ladder, Ref, Comparator, etc.
3. Word Length Storage, etc.	10	.09	34	
4. Compressor	21	.15	80	
5. Control Counters	50	.46	236	
6. MUX (64 Channels) 32 Analog 32 Digital	76	.50	173	6 MEM 2009, 32 Transistors, 32 Resistors (.35 Boards)
7. Housekeeping	49	.35	175	
Subtotal	267	2.53	1,378	
8. 6 Accumulators	156	1.08	612	
Total	423	3.61	1,990	
	Power and Weight Reductions for Removing			
1 Accumulator	26	.18	102	
8 Analog Channels	12	.12	16	1-MEM 2009 8 Resistors
8 Digital Channels	8	.05	8	
1 DOR (8 Bits)	9	.06	36	

3.4.2 I/O - CCU Interface

There are two parts to the interface between the I/O and the CCU: Multiplexor Data Control; and Discrete Output Register control. There are 5 8-bit Discrete Output Registers (DOR's) providing 40 discrete outputs (see Figure 3.4-3). A DOR is individually selected and any number of the 8 bits of that DOR can be set or cleared at one time by a set or clear discrete instruction. It is also possible to set or clear the same bit positions in more than one register at a time. Five register lines and 8 bit lines are presented to the DOR section by the CCU, along with a Set and a Clear line and an Action line. The Set and Clear lines determine whether a register or registers are to be set or cleared, the bit lines determine which of the 8 bit positions are to be set or cleared, and the Action line causes the action to take place. The discrete outputs are available for connection to the CCU, external systems, or the I/O section itself.

The Multiplexor Data Control interface has 18 lines as described in Section 3.2. The CCU specifies a data channel (and type) with the 12 address lines (1, 2, 4, 8, X, Y, $\overline{1}$, $\overline{2}$, $\overline{4}$, $\overline{8}$, $\overline{X}$, and $\overline{Y}$), the word length with two lines (one to four bytes) and brings the Command line to a "one" requesting the data from that channel. The I/O performs those processes necessary to acquire the data. When the data is available the I/O brings the Data Ready line to a "one." The CCU then reads out the data with the Data Clock. When the correct number of bytes has been read, the I/O drops the Data Ready line ending the data exchange. The Command line will drop and the I/O will wait for the next data request.

The response time of the I/O section in acquiring data is a function of the type of data and, except for analog data, the number of bytes to be

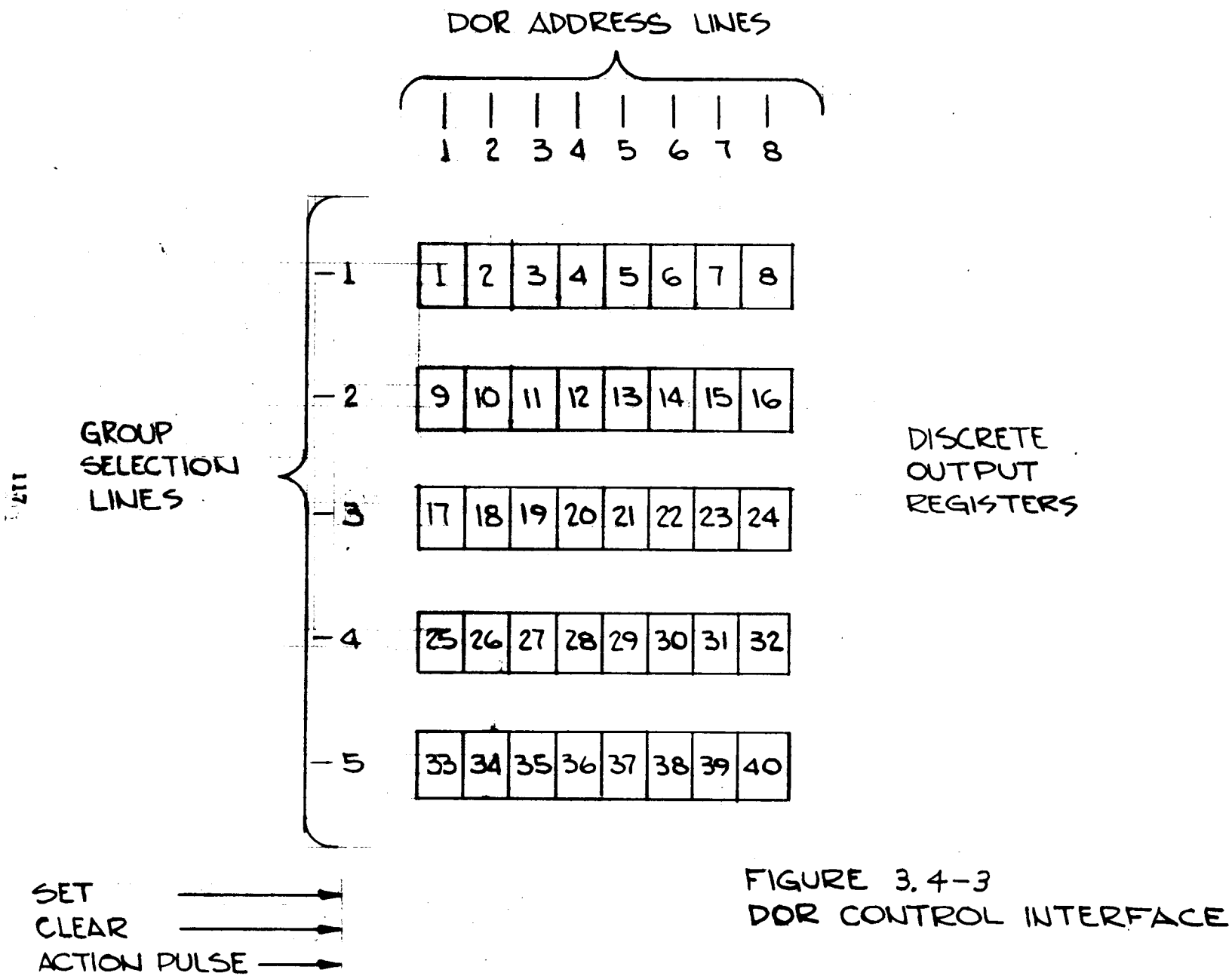
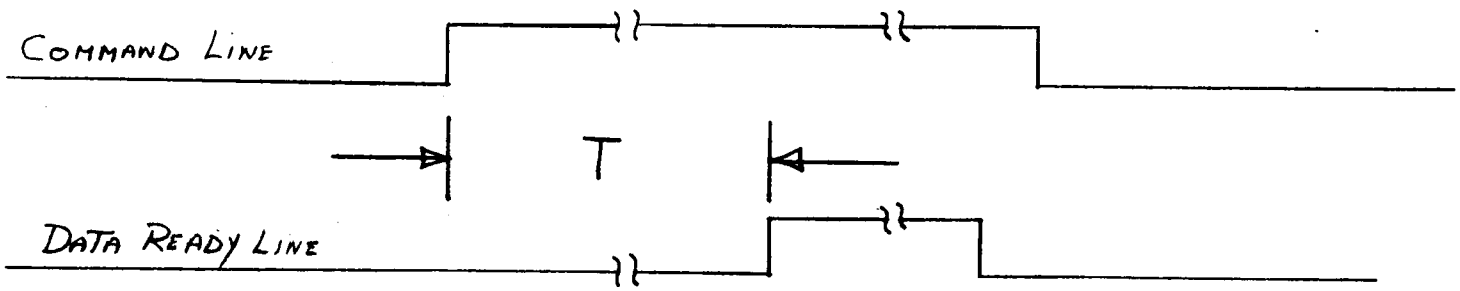


FIGURE 3.4-3
DOR CONTROL INTERFACE



DATA TYPE	T (μsec)
ANALOG	112
ACCUMULATOR	8 TO 25
DIGITAL	2

FIGURE 3.4-4

DATA READY RESPONSE TIME

gathered. Response time, measured from the Central Control's Fetch Command to the I/O's response with a Data Ready is shown in Figure 3.4-4. Following the Data Ready signal, the Central Control shifts the data out in serial bursts of one byte length at an 800 KHz bit rate. One byte is transferred per 10 sec machine cycle.

The I/O interprets the two work length bits according to Figure 3.4-5 below. The data type (analog, digital, and compressor or accumulator) is defined by the channel address lines and channels are assignable in modules of 8 channels.

W. L. Bits	Analog	Digital	Compression
00	2 Byte (8MSB of 10)	4 Byte	--
01	--	1 Byte	--
10	2 Byte (8LSB of 10)	2 Byte	2 Byte (8LSB of 9)
11	3 Byte (10 bits + 2 Filler)	3 Byte	3 Byte (9 bits + 3 Filler)

Figure 3.4-5

Number of Bytes by Data Type

3.4.3 Logic Description

3.4.3.1 Discrete Output Register

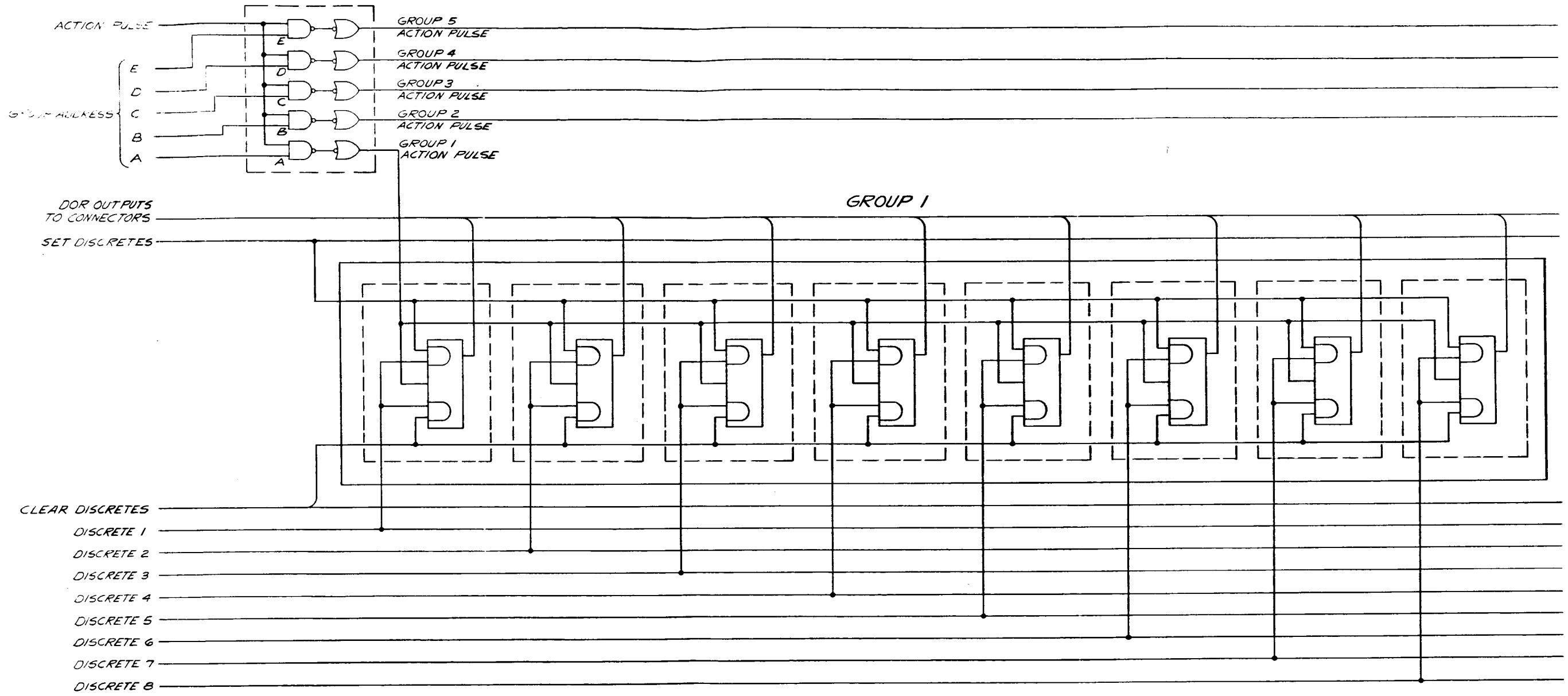
The discrete Output Register consists of up to five 8-bit storage registers. Data storage is parallel in, parallel out. The Discrete Output Register interface between the Multiplexer and the Central Control consists of 16 signal lines. Five lines select the eight bit register (s) being addressed; eight lines select any of the eight within that group; one line instructs set; one line instructs clear; and one line initiates the action to be taken. It is possible in this way to change any one storage bit or group of storage bits without affecting the remaining bits.

Each discrete is stored in a LPDT L 9040 flip flop. As can be seen from Figure 3.4-6, storage is accomplished by strobing eight "trigger" inputs simultaneously while selectively enabling eigher the "set" or "reset" steering inputs by coincidence of a discrete address signal and a set/clear signal.

The states of the discretes are outputed from the I/O through a connector. These signals are user assigned to fulfill the requirements of any given mission. They may be used to control experiment sensor modes or wired back into the DPS for program or I/O control (such as for accumulator input gating.)

3.4.3.2 Channel Selection

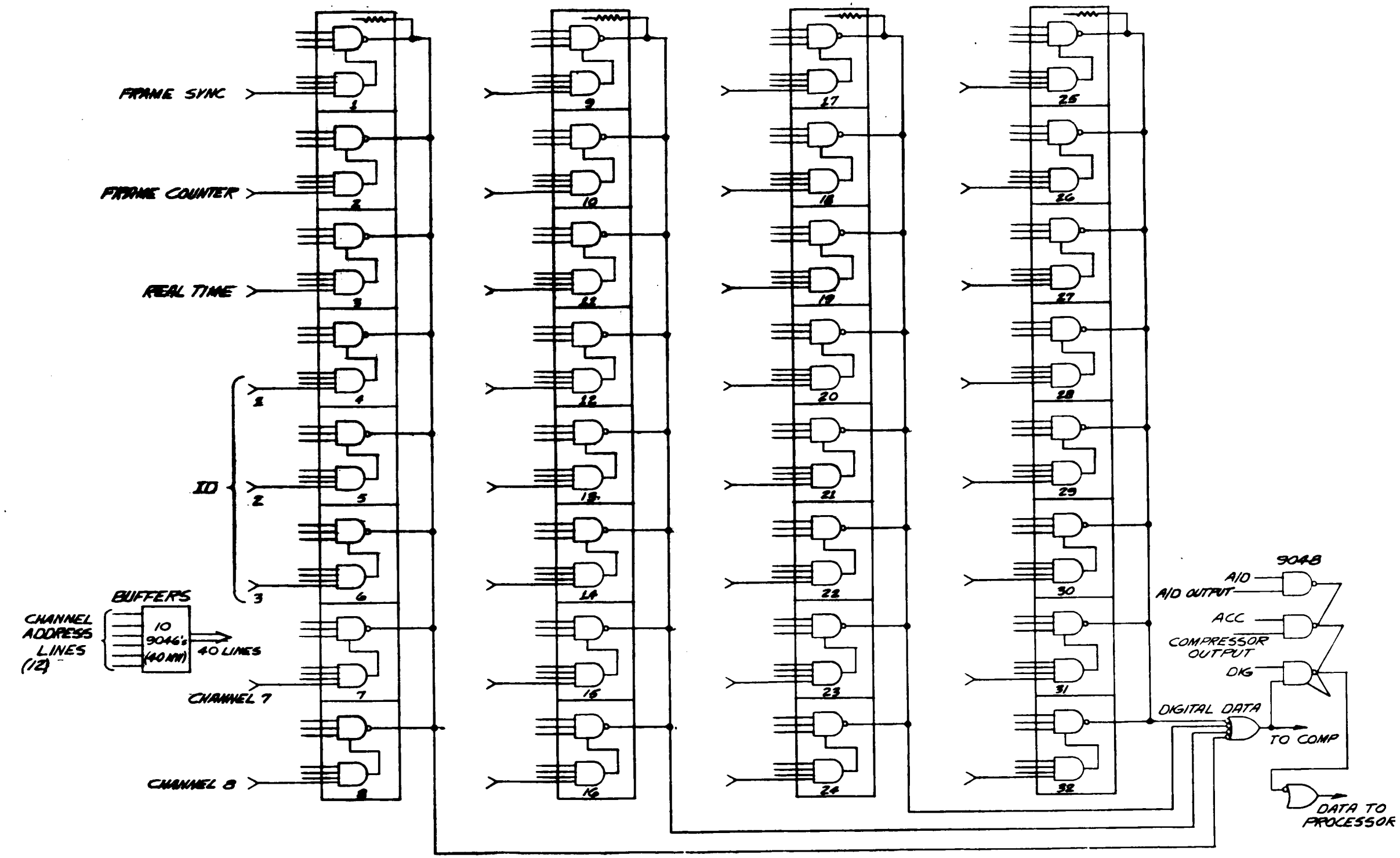
Data channel selection is accomplished by decoding the six "channel address" lines with seven input NAND gates (a 3 input NAND plus a 4 input expander), see Figures 3.4-7A, and 3.4-7B. For digital channels, the 7th input is the data source; for analog channels the 7th input is unused, and the gate drives an analog switch. The basic design allows for 32 digital and 32 analog channels; there is a further division into a total of eight groups of eight channels each. Each of the three types of data sources must be



COMPONENT USAGE

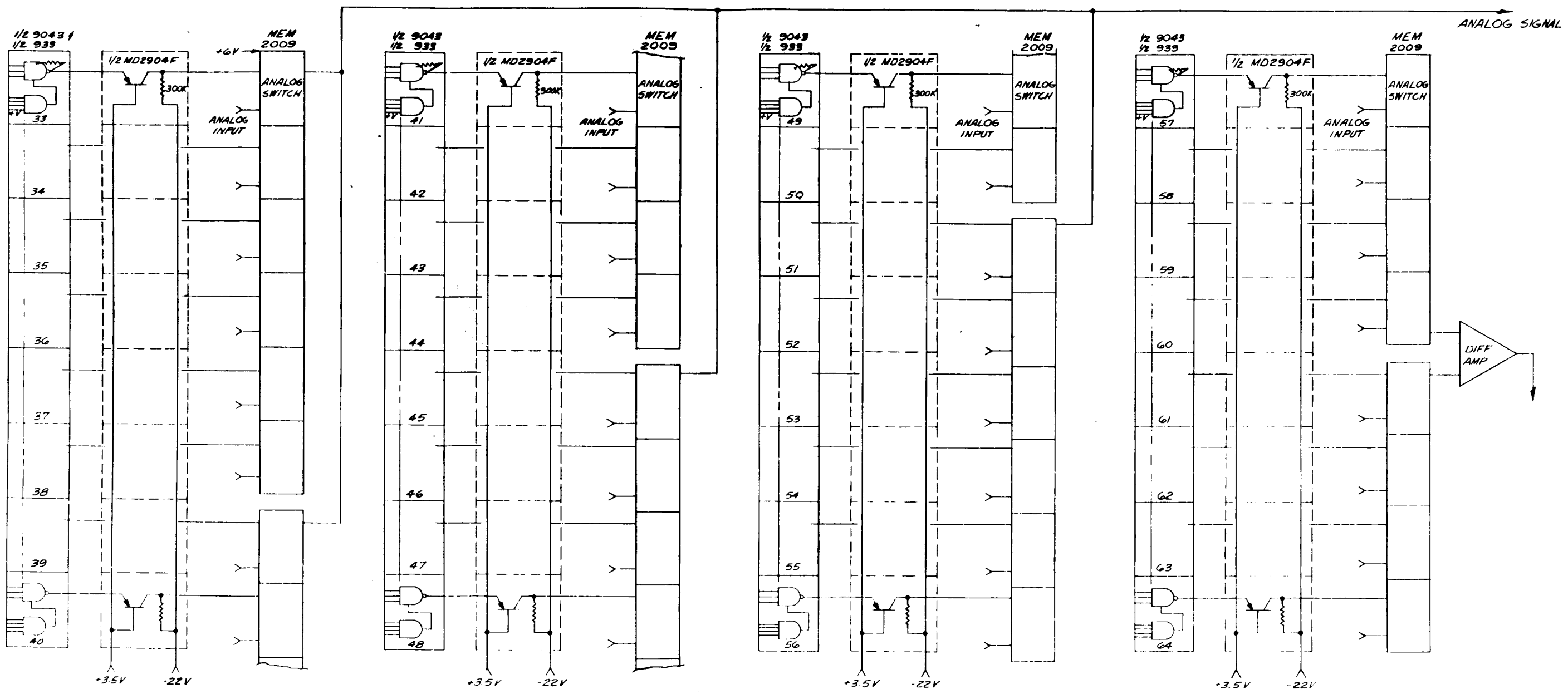
5	9046	20 MW
40	9040	160 MW
45		180 MW

FIGURE 3.4-6 DISCRETE STORAGE



DIGITAL CHANNEL SELECTION

FIGURE 3.4-7A

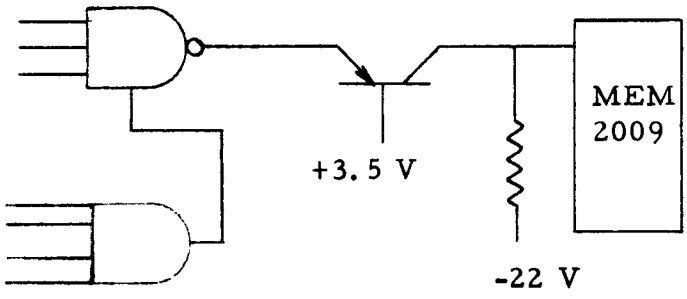


ANALOG CHANNEL SELECTION

FIGURE 3.4-7B

assigned to separate channel groups - no group of eight may have two types of data sources. The number of channels assigned to any data type is expandable in groups of eight. Though 64 channels are provided, individual components may be removed when fewer channels are channels may be duplexed. Also, the analog channels can be changed to digital channels by interconnections available at the connector, and components can be removed to save power. Any of the 64 primary channels may be subcommutated, making almost limitless the total number of data sources that can be accommodated.

Analog switching is accomplished by MEM 2009 analog switch arrays driven by a LPDTμL NAND gate and a PNP transistor level shifter network.

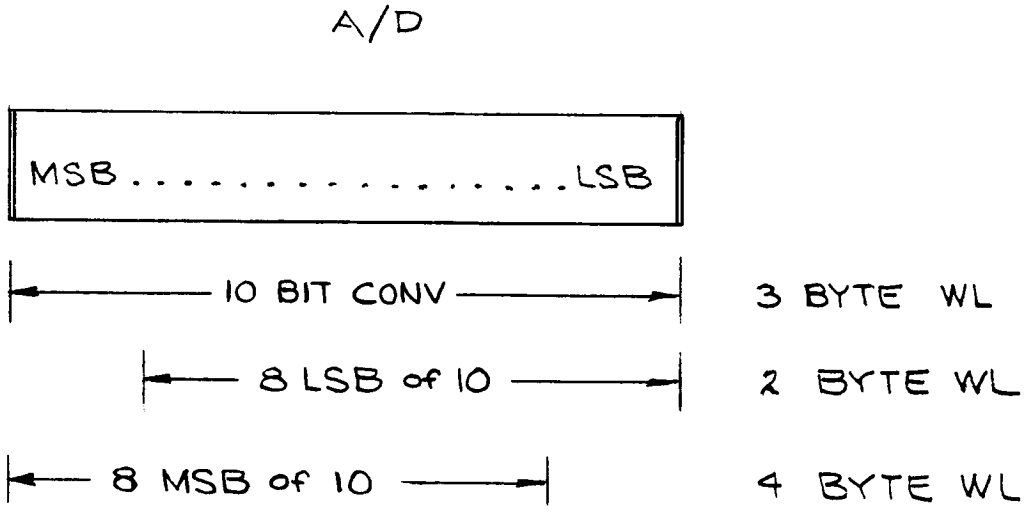


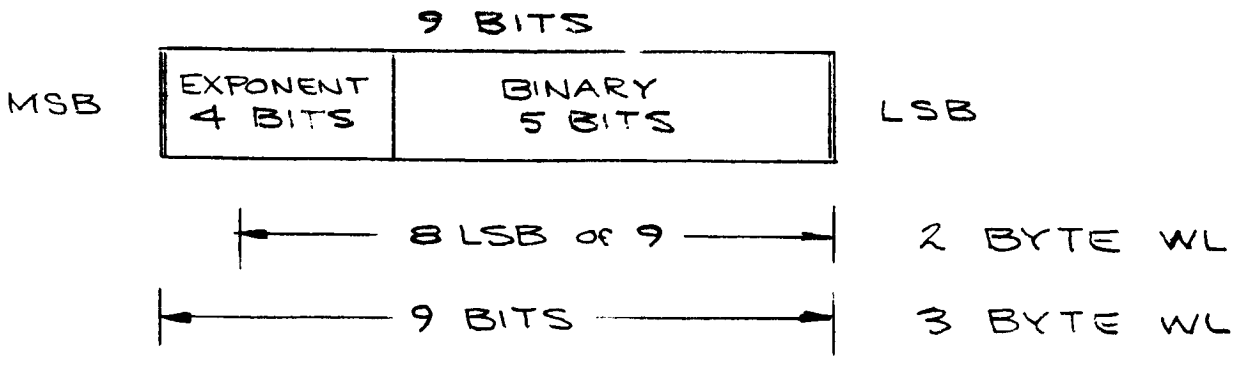
Typical +5V to -22V Level Shifter

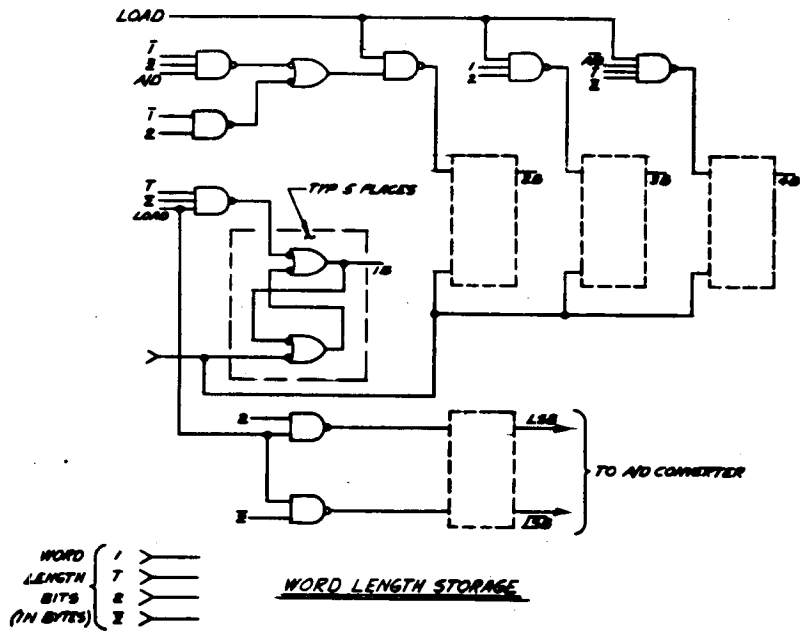
Though not specified, the present design includes a capability for eight differential analog inputs.

3.4.3.3 Word Length Storage and Operation Decoding

Patching at the adaptive interface establishes the data type - channel number relationship for the requirements of each flight. By patching and decoding the multiplexor address signals X, Y, and 8, the "data type" is specified for each group of eight channels. See Figure 3.4-8. The three signal lines "A/D," "ACC," and "DIG" are used to control logic actions. As discussed in Section 3.4.2 the word length (WL) bits are interpreted using the "data type" signals. The interpretation is illustrated below:

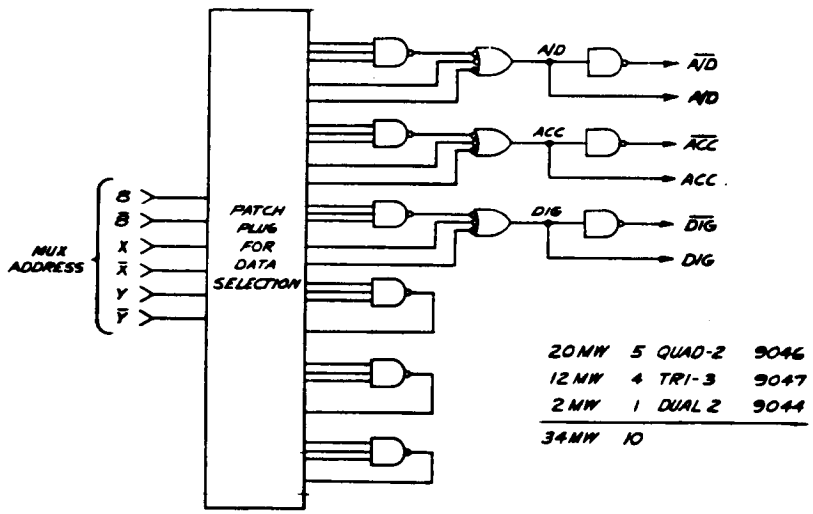






WORD LENGTH STORAGE

FIGURE 3.4-8A



20 MW	5	QUAD-2	9046
12 MW	4	TRI-3	9047
2 MW	1	DUAL 2	9044
34 MW	10		

OP TYPE LEVEL DECODING

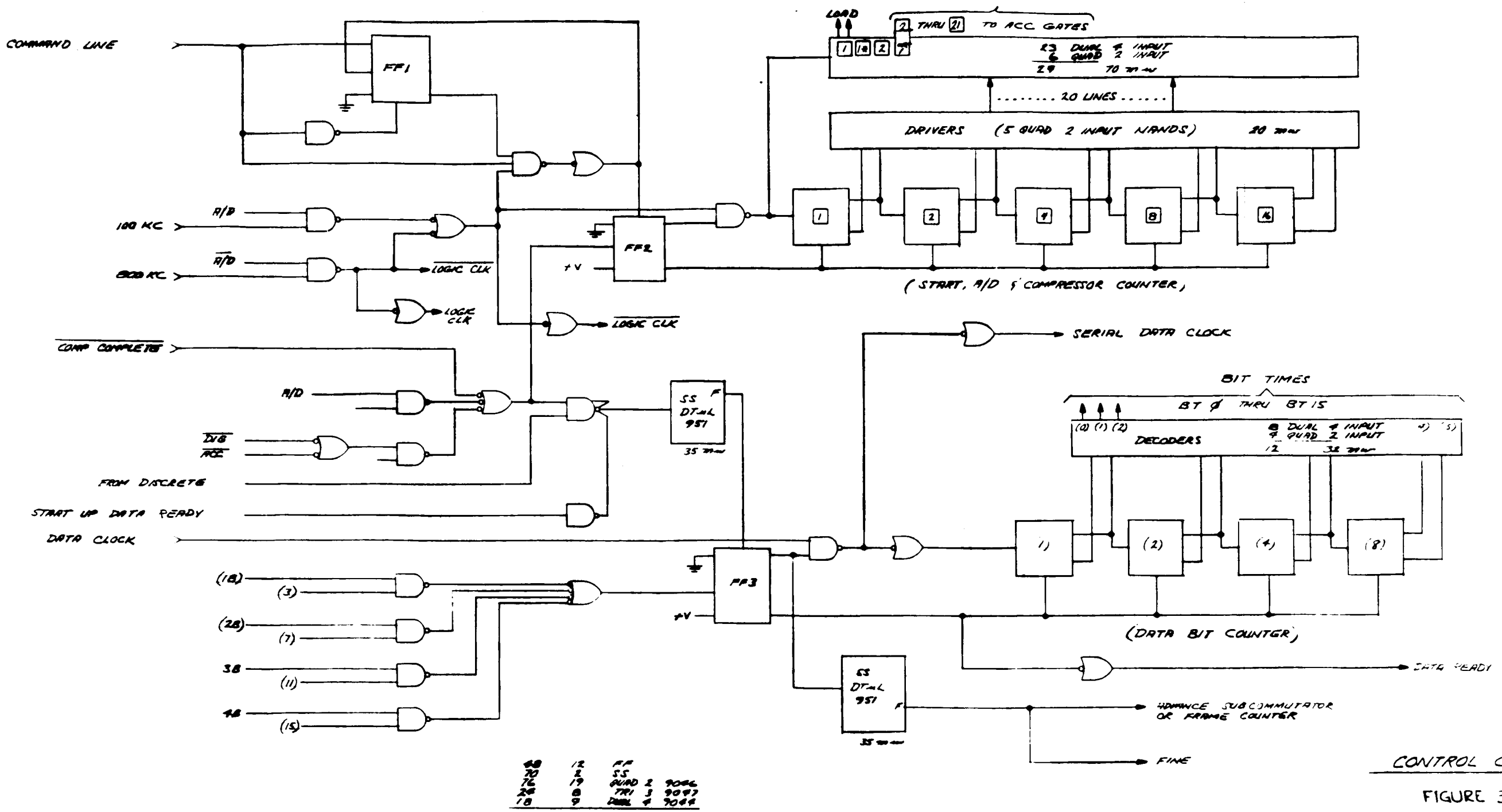
FIGURE 3.4-8B

3.4.3.4 Control Counters

The heart of the multiplexor logic is the Control Counter section, see Figure 3.4-9. Here the multiplexor commands are received and executed. There are two series of events which take place. The first series acquires digital data to be sent to the processor. The second series sends the data.

The first series begins with a "0" to "1" transition of the Command line. This enables a counter to generate a series of N pulses on N lines; the number N is determined by the type of data being acquired. Pulse number one, [1], loads the channel address and the word length for all types of data. For 20-bit accumulator data, and analog data pulse [1*] initiates the compression, [2] initiates the analog-to-digital conversion process; for serial digital data, [2*] sets the Data Ready flip flop. Lines [2] through [21] strobe the accumulator gates providing a parallel to serial conversion of the accumulator data. All accumulators are strobed simultaneously, but only the data from the addressed channel is passed to the compressor. A Compression Complete signal, which sets the Data Ready flip flop, is generated by the compressor. Signal line [13] sets the Data Ready flip flop at the conclusion of an analog-to-digital conversion; all A/D conversions are done to 10 bits.

The second series of events begins with the setting of the Data Ready flip flop. This informs the Central Control (via the Data Ready line) that the Multiplexor has data available and is waiting for the Data Clock to transfer the data to the Central Control. The Data Bit Counter counts the Data Clock pulses; provides strobe pulses to the compressor, A/D converter, and serial digital data sources; and resets the Data Ready flip flop when the correct number of bytes have been transferred to the Central Control.



48	12	FF
70	2	SS
76	19	QUAD 2 9046
24	8	TRI 3 9093
18	9	DUAL 4 9044

CONTROL COUNTERS

FIGURE 3.4-9

128-2

128-1

3.4.3.5 Frame Sync and ID Synthesis

The Frame sync and ID subsection (Figure 3.4-10) provides the following capability:

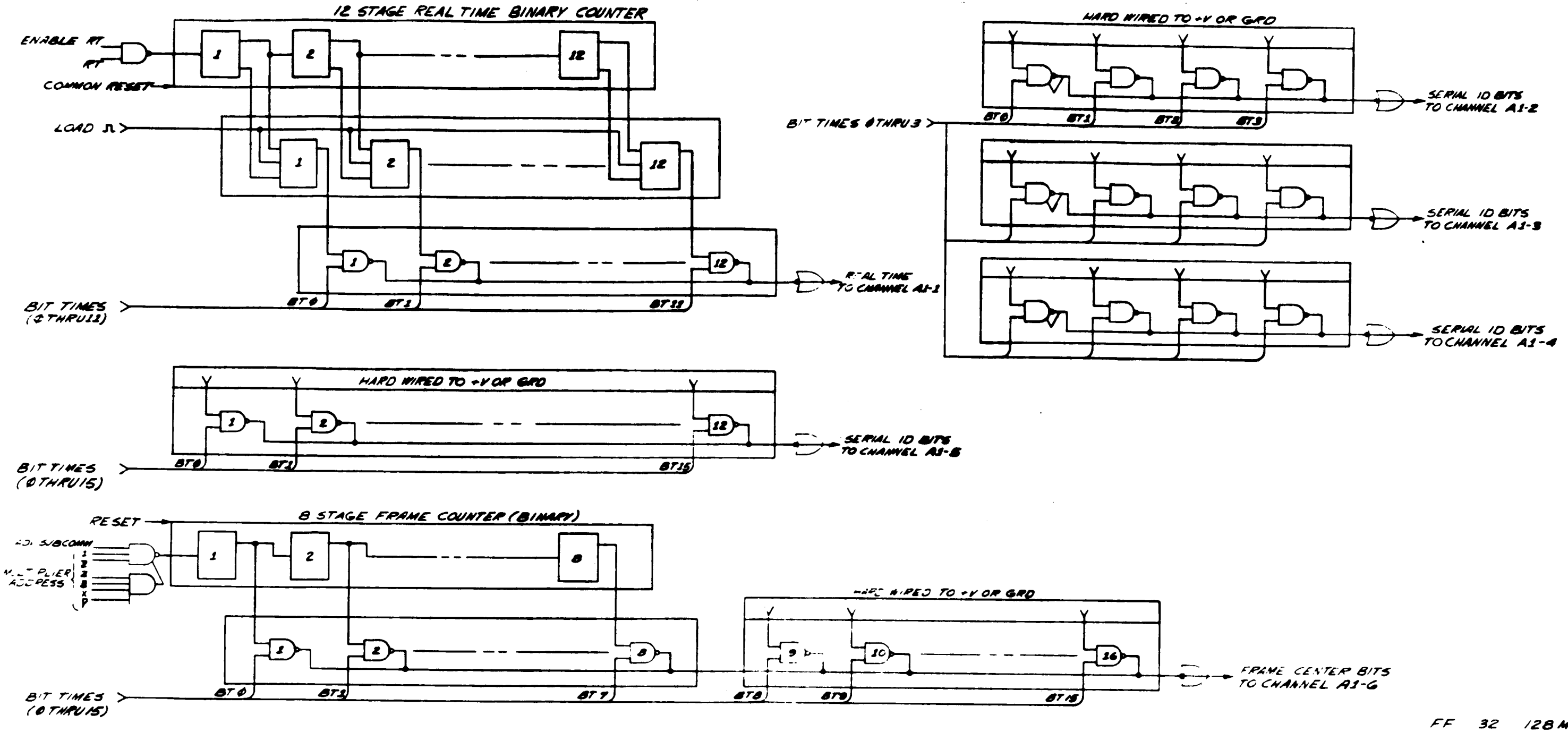
- 12 bits real time counter
- 24-bits patchable frame sync
- 8 bits frame counter
- 12 bits (three groups of four bits) patchable Id bits

The readout of all three data types is identical; the data bit count lines strobe the many data gates simultaneously. The addressed data channel and/or data type being the only one passed to the Central Control over the Data Line

3.4.3.6 Analog-to-Digital Converter

The analog-to-digital conversion process (Figure 3.4-11) uses the standard resistive ladder successive approximation technique. Analog switch and sequencing is accomplished by an 810 device. The 810 stores the digital data, but its readout rate is significantly lower than the LPDTμL (100 kc vs. 2 mc). In order that the multiplexor to processor data transfer always occurs at the fastest rate, the 810 real-time output (A/D Out) is stored in a LPDTμL shift register. Digital data readout is accomplished by two input NAND gates, strobed by the Data Bit Count signals. Because of the three A/D conversion requirements (10 bits, 8 MSB, and 8 LSB) two sets of gates are required. The control lines, LSB and $\overline{\text{LSB}}$, come from FF1 Figure 3.4-8, which has interpreted the data request.

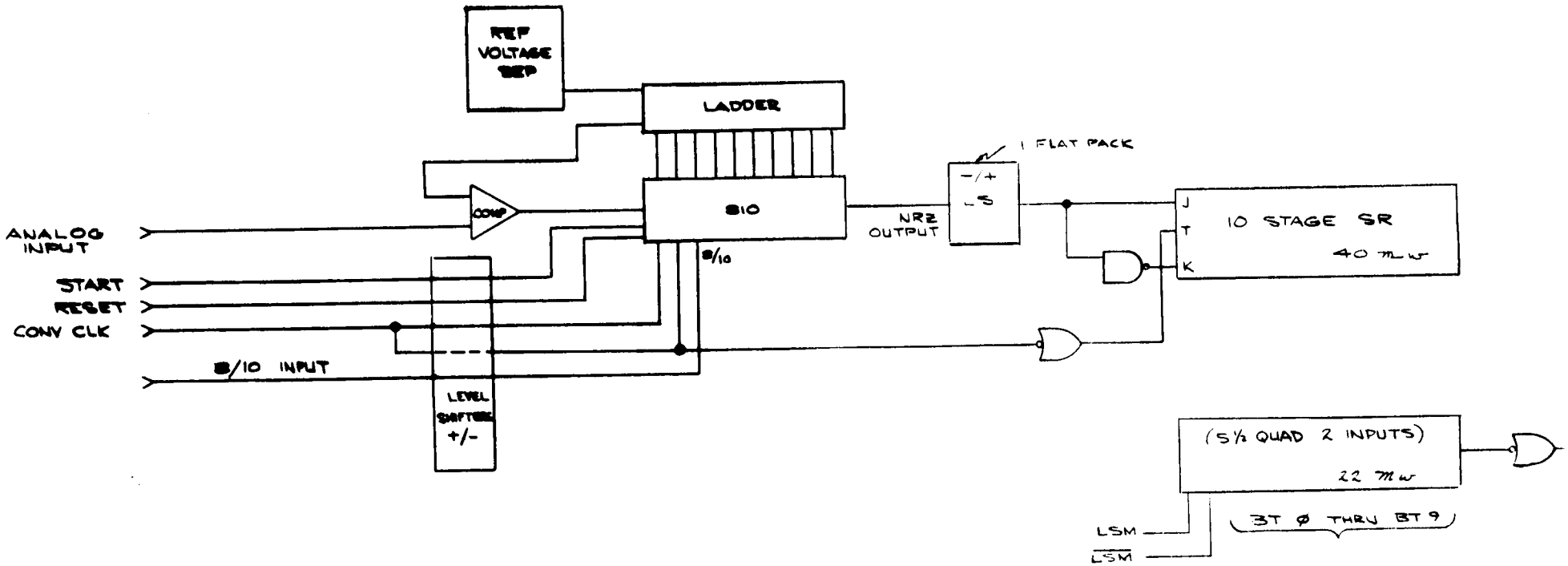
Details of the A/D converter operation and test data are given in Section 3.5.



FF	32	128 MW
QUAD 2	16	64
QUAL 4	1	2
	49	174

FRAME SYNC & ID SYNTHESIS

FIGURE 3.4-10



A/D CONVERTER

FIGURE 3.4-11

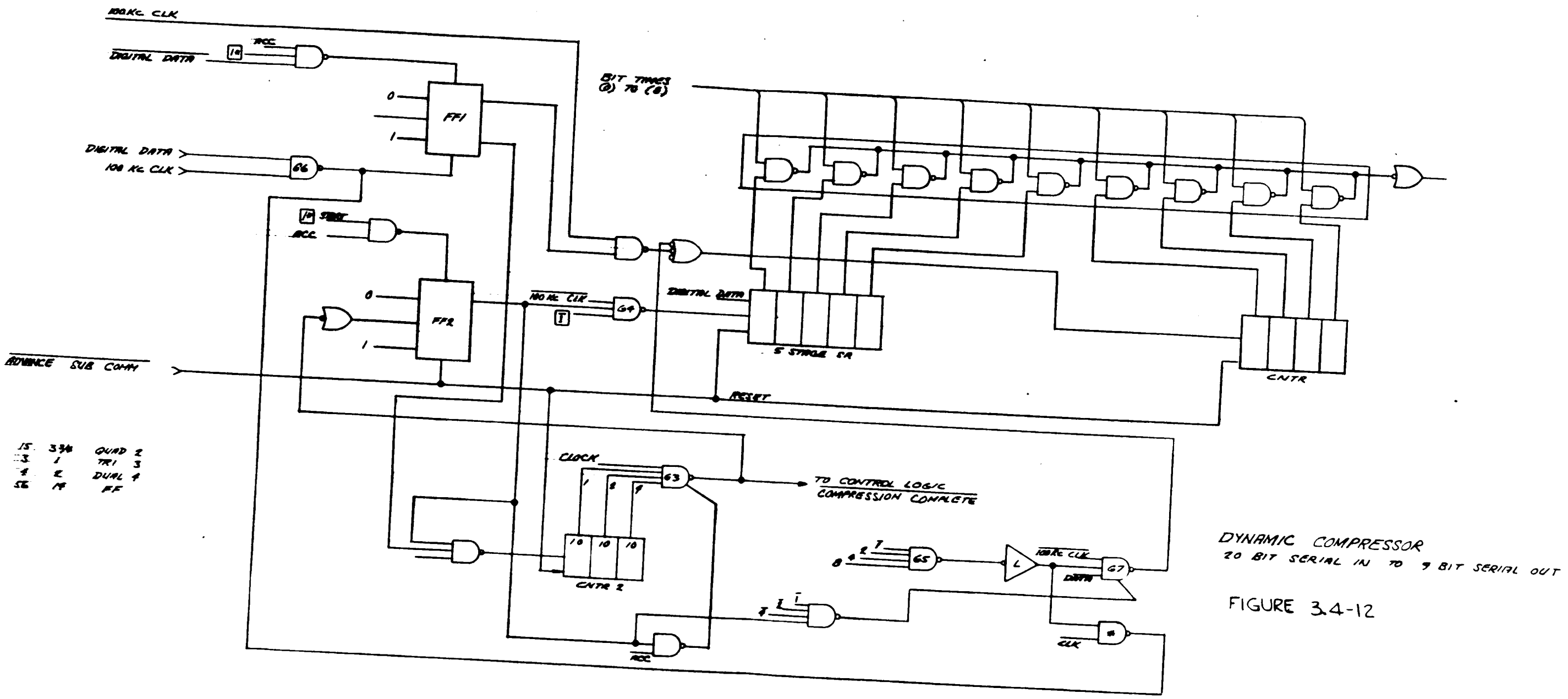
3.4.3.7 Compressor

The compressor (Figure 3.4-12) accepts 20 bits of binary data (2^0 through 2^{19}), generating a nine-bit output. Four of the nine bits represent an exponential multiplier; five of the bits represent a binary number. Details of the mathematical interpretation of the compressor output are given in Section 4.0. Accumulator data is shifted into the compressor, MSB first; a down counter, starting at "15" records the number of shift pulses required to bring a data "one" into the compressor. When a "one" is received, the down counter is turned off; it now contains the exponential multiplier. The five data bits following the first date "one" (or, if no "one" is received in the first 15 bits, the five LSB's) are stored in a shift register. At the completion of the storage, a compression Complete pulse sets the Data Ready flip-flop. At the completion of the data readout, the compressor is cleared and made ready for the next accumulator channel data request.

Figure 3.4-13 shows the timing sequence for two situations. Situation 1 shows the sequence that occurs if there is no "one" in the 15 MSB's, bit 20 through bit 6. Situation 2 shows the sequence when the most significant "one" is located in data bit 16.

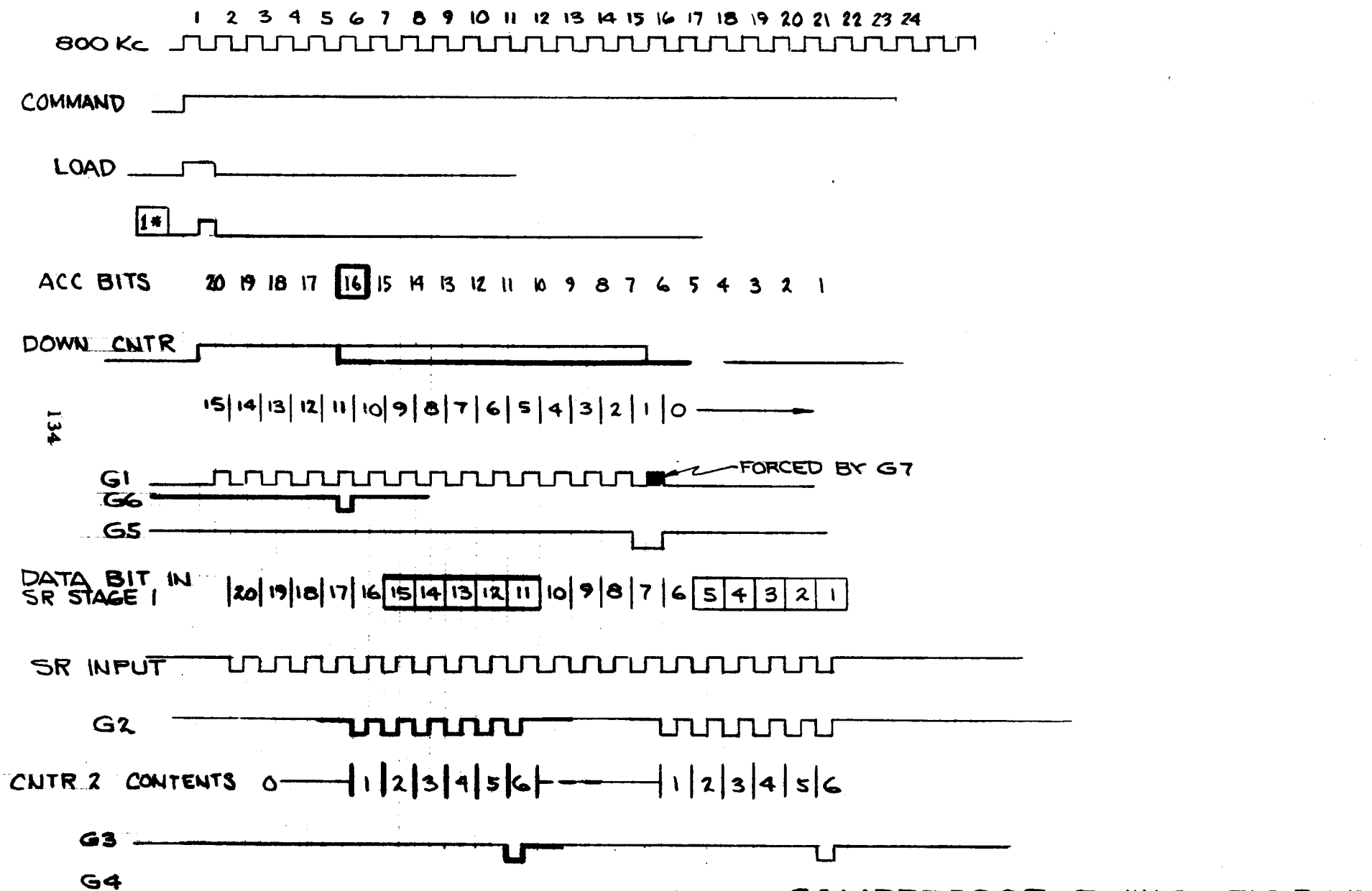
Situation 1

The compression process begins with 1* trying to set both FF1 and FF2, if the accumulator being sampled has a "one" in the MSB, only FF2 is set. The coincidence of a data "1" and the signal Logic Clock resets FF1. The first "one" is not stored, the next five bits are stored; therefore, six Store pulses are required after FF1 is reset. FF1 is reset during the Bit 6 time (no data had been received previously. If Bit 6 had been a "1" and that had caused FF1 to Reset, a remainder of "1" in the counter would



DYNAMIC COMPRESSOR
20 BIT SERIAL IN TO 9 BIT SERIAL OUT

FIGURE 3.4-12



COMPRESSOR TIMING FIG. 3.4-13

have been correct, and the next five bits would be stored in the shift register. However, in this case, Bit 6 is a "zero", therefore, the down counter must receive one more count. Gate G7 provides this pulse. Gate 3 provides the reset for FF2, ending the compression process and setting the Data Ready flip flop.

Situation 2

Coincidence of a data "one" and Logic Clock at Accumulator bit 16 resets FF1, enabling Gate G2. Counter 2 records six clock cycles, storing accumulator bits 15, 14, 13, 12, and 11 in the shift register. Gate G3 resets FF2 and sets the Data Ready flip flop signaling the end of the compression process. The down counter contains 1110.

Readout

Readout is done through a series of two input NAND gates sequentially strobed by the data bit count signals, in exactly the same way that serial digital and converted analog voltage data readout is done.

3.4.3.8 Command Uplink Input

The DPS has the capability, through the bootstrap loader and two load discretes, for loading new programs into memory. One discrete, not program interpretable, forces the machine to the starting location of the bootstrap loader. The other discrete, called load, is a branch condition for control of the load program and is interpreted by Branch instructions in the loader. The load routine includes a Fetch instruction which addresses a fixed multiplexor digital channel in the 8-channel group which contains the message header hardware, and asks for one

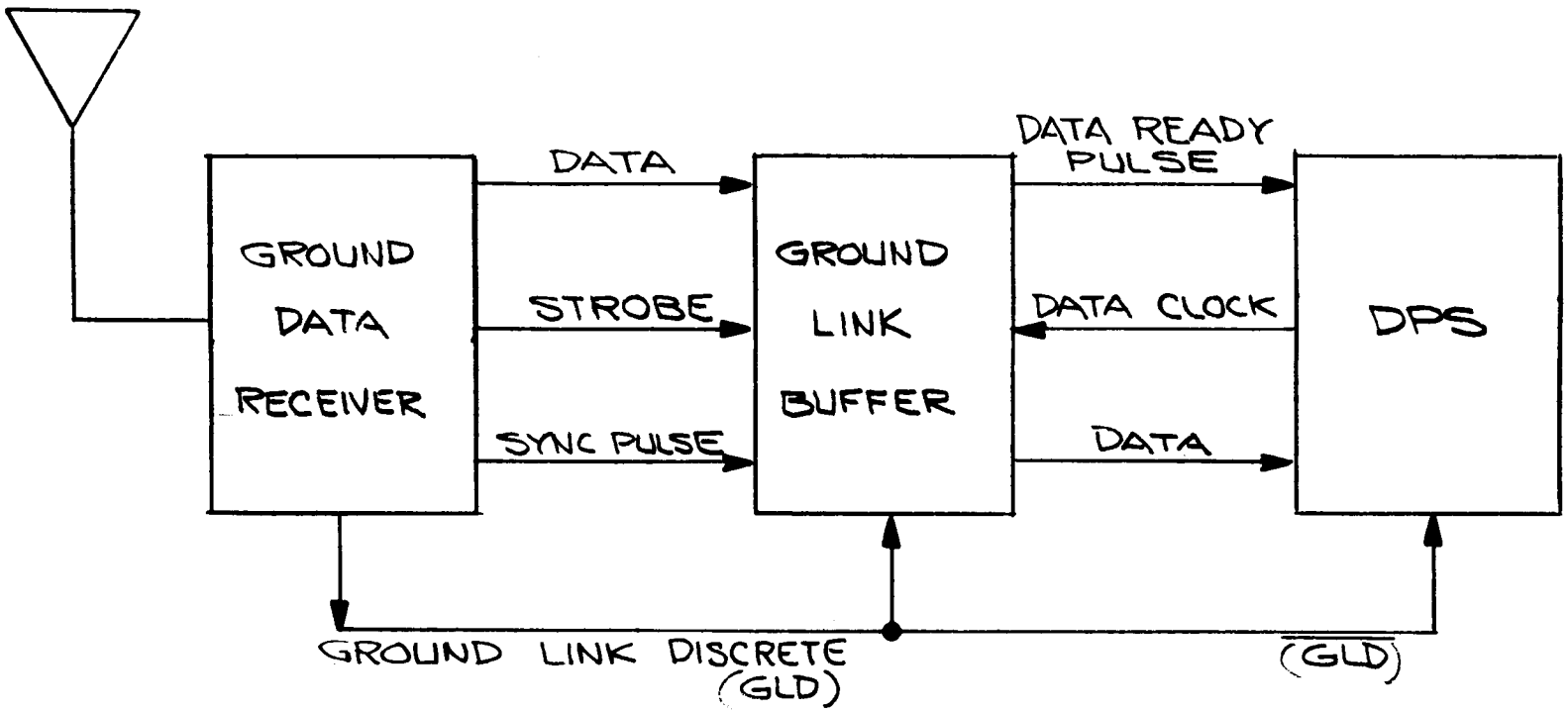
byte of data each execution. The time interval between gathering bytes of data from this channel is 100 μ seconds or less. A one byte shift register at the input to this channel would permit input bit rates of 10K bps; two one-byte registers would permit input bit rates of 40K bps.

The capabilities of the command system were not determined during this study nor was a specific interface worked out. The following discusses one approach to this interface with the S^3 DPS, and assumes certain capabilities within the command system. The interface consists of a Ground Link Buffer to interface with the S^3 DPS through a digital channel and assumes certain interfaces with the command system (see Figure 3.4-14). Because this area was not explored with NASA, the component count associated with this assumed interface is not included in the S^3 DPS.

It is assumed that the receiver or command system would provide the following signals:

- Serial Data
- Serial Data Strobes
- A Sync Pulse indicating the beginning of Data
- A Force bootstrap discrete
- A "Load" discrete (a branch condition)

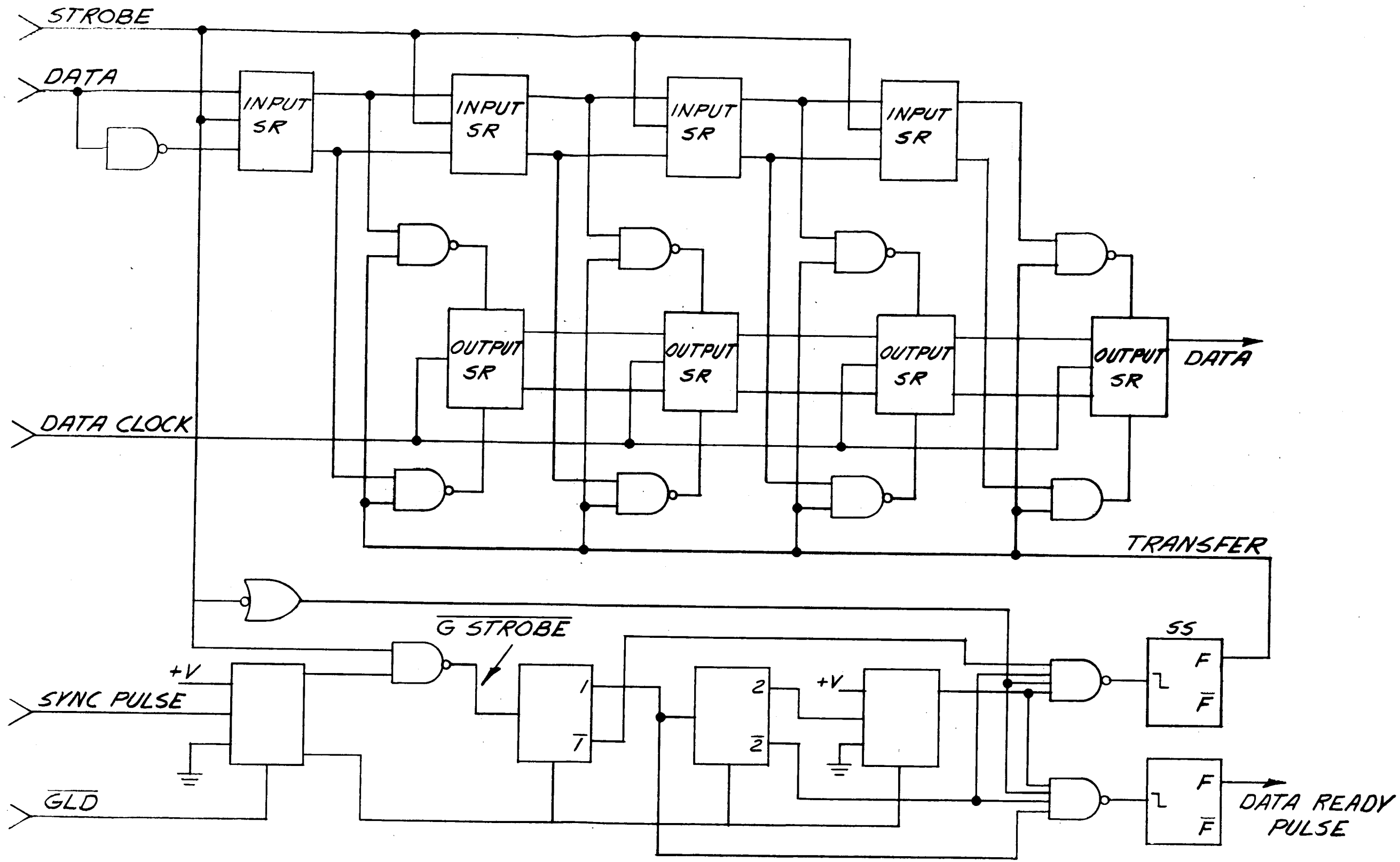
The last two signals could be applied simultaneously. The last signal should be applied until the last bit has been presented to the input; changing it terminates the bootstrap load. After these signals have been



BLOCK DIAGRAM - GROUND TO DPS DATA LINK

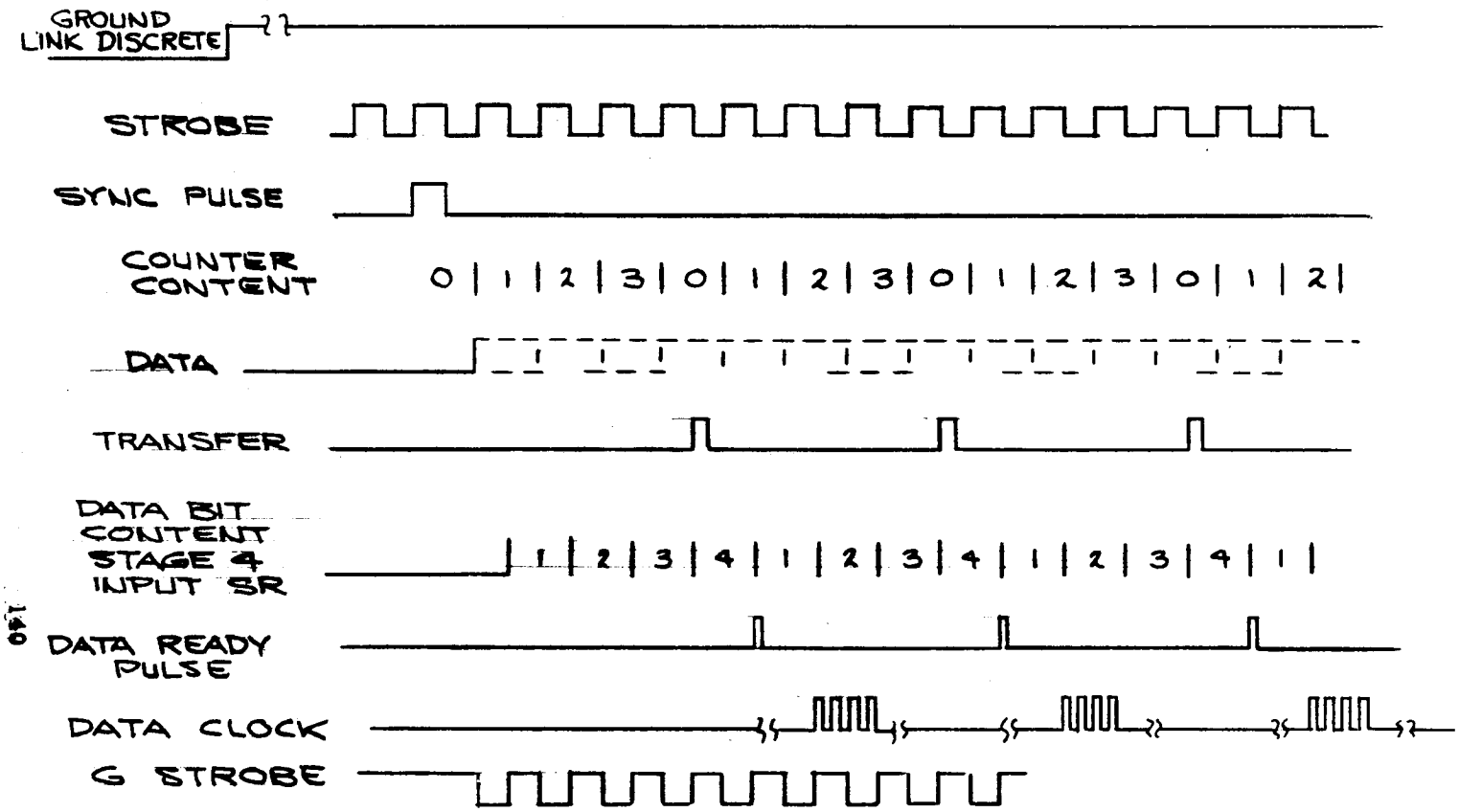
FIGURE 3.4-14

applied 300 μ seconds should be allowed for initialization and executing the loader to the point of issuing the first Fetch command to the I/O section. Following this, the Sync pulse can enable a 4-bit counter and data can be strobed into a 4-bit shift register. The data strobe will also increment the 4-bit counter. Each time 4 bits have been shifted into the input shift register, the 4-bit byte is transferred to a 4-bit output shift register and the Data Ready signal is set. The data will then be shifted at the 800KHz rate to the Central Control for input into the proper place in memory. This operation proceeds until all bytes of the new program have been entered and the "load" discrete is changed to signify the end of the loading operation. The input bit rate in this scheme is 40K bps. Figure 3.4-15 illustrates the logic which would perform this function with the assumed receiver interfaces. Figure 3.4-16 is a timing diagram for this logic.



GROUND LINK BUFFER LOGIC DIAGRAM

FIGURE 3.4-15



GROUND LINK BUFFER TIMING DIAGRAM

FIGURE 3.4-16

3.5 Circuit Technology

The circuit requirements for the S^3 system include: logic circuits for the performance of general logic functions throughout the system; input-output circuits to perform the functions of multiplexing of analog and digital data, analog-to-digital conversion, and load driving; interface circuits to provide proper signal matching between MOSFET and bipolar transistor circuits; and an oscillator to provide stable timing for both the system and the data synchronized clock. In the following paragraphs, the technology chosen for each of these circuit requirements is discussed and design specifications are presented where appropriate. The memory circuits are described separately in the memory section.

In selecting the technology mix a number of design trade-offs were conducted. In the Memory and Central Control sections the trade-offs were rather obvious and power and weight optimization quickly became a matter of optimized design. The technology selection in the I/O section, however, required several complete designs with various types and mixes of technologies to arrive at the mix permitting both system weight and power to be optimized. In the end, the use of MEM-5015 MOSFET's for multiplexing was dropped in favor of a hybrid multiplexing implementation for reducing power consumption at the expense of MIB area. While MIB area implies added weight, only the difference in component weights entered into the system weight - there was no difference in delta packs. These trade-offs are discussed in Section 4.0.

3.5.1 Logic Circuits

The Fairchild LPDTuL (Low-Power Diode-Transistor Micro-Logic) integrated circuit family has been chosen for general logic applications in this system because of the low power required, the compatibility of the family with other voltage-mode families (DTL in general), and the logic design efficiency that can be obtained with the family. Because of the

compatibility of LPDTuL with DTuL (the Fairchild designation for its higher-power, higher-speed DTL family), the latter is used for those applications requiring either high fan-out or load-driving capability. Careful choice of the supply voltages and the use of a low-power matching circuit also allows interfacing LPDTuL with MOSFET circuits. The interfacing circuits are described in Section 3.5.2.4.

The LPDTuL circuit family characteristics are listed in Table 3.5-1. The design ground rules that have been used in applying the circuits in the S^3.

The operating temperature range over which the design specifications for the LPDTuL family applies is the standard military range of -55°C to 125°C. The expected operating temperature of the circuits in the S^3 range between -10°C and +65°C. The narrower temperature range of the S^3 environment would allow slight improvement in the application ground rules. Since the amount of improvement would not significantly change the total system circuit requirements, the full temperature range specifications have been chosen to apply. This effectively improves the operating margins of the circuits which will contribute to the system reliability and ability to withstand the radiation environment.

The basic LPDTuL circuit is illustrated in Figure 3.5-1. The configuration is a standard DTL configuration using high-valued resistors to obtain low power requirements at the expense of circuit speed. Since the S^3 design does not require low propagation delays in the logic, and since the system power requirement must be kept low, the speed-power tradeoff implied in the choice of the LPDTuL family is appropriate.

The basic circuit allows the use of either a resistor pull-up or an emitter follower pull-up in the output. Provision is made to allow

TABLE 3.5-1

LPDTμL CHARACTERISTICS

TYPE NO.	FUNCTION	AVG POWER*	FAN OUT
9040	Clocked Flip-Flop	4 mw	10
9041	dual 3-input gate (programmed output)	2 mw	10; 7 w/pull-up
9042	Dual 3-input Gate (EF Pull-up)	2 mw	10
9043	3 & 4 Input Gate (With Extender)	2 mw	10; 7 w/pull-up
9044	Dual 4-input Gate (w/extender, EF Output)	2 mw	10
9046	Quad 2-Input Gate (EF Output)	4 mw	10
9047	Triple 3-input Gate (EF output)	3 mw	10
9048	Quad 2-input Gate (no Pull-up)		7 w/separate pull-up
9049	Triple 3-Input Gate (no Pull-up)		7 w/separate pull-up
9944	(DTuL) Dual 4-Input power Gate (EF Output)	20 mw	50

* For reference purposes only. The actual dissipation figures depend on the output pull-up that is used and the ratio of "on" time to "off" time. The figures used for the logic circuits are summarized below.

OFF		ON		
Max.	Avg.	Max.	Avg.	
0.785	0.550	1.8	1.2	No pull-up
**	**	2.2	1.63	R pull-up
**	**	**	**	EF pull-up

** Transition dissipation only, depends on load C and other factors.

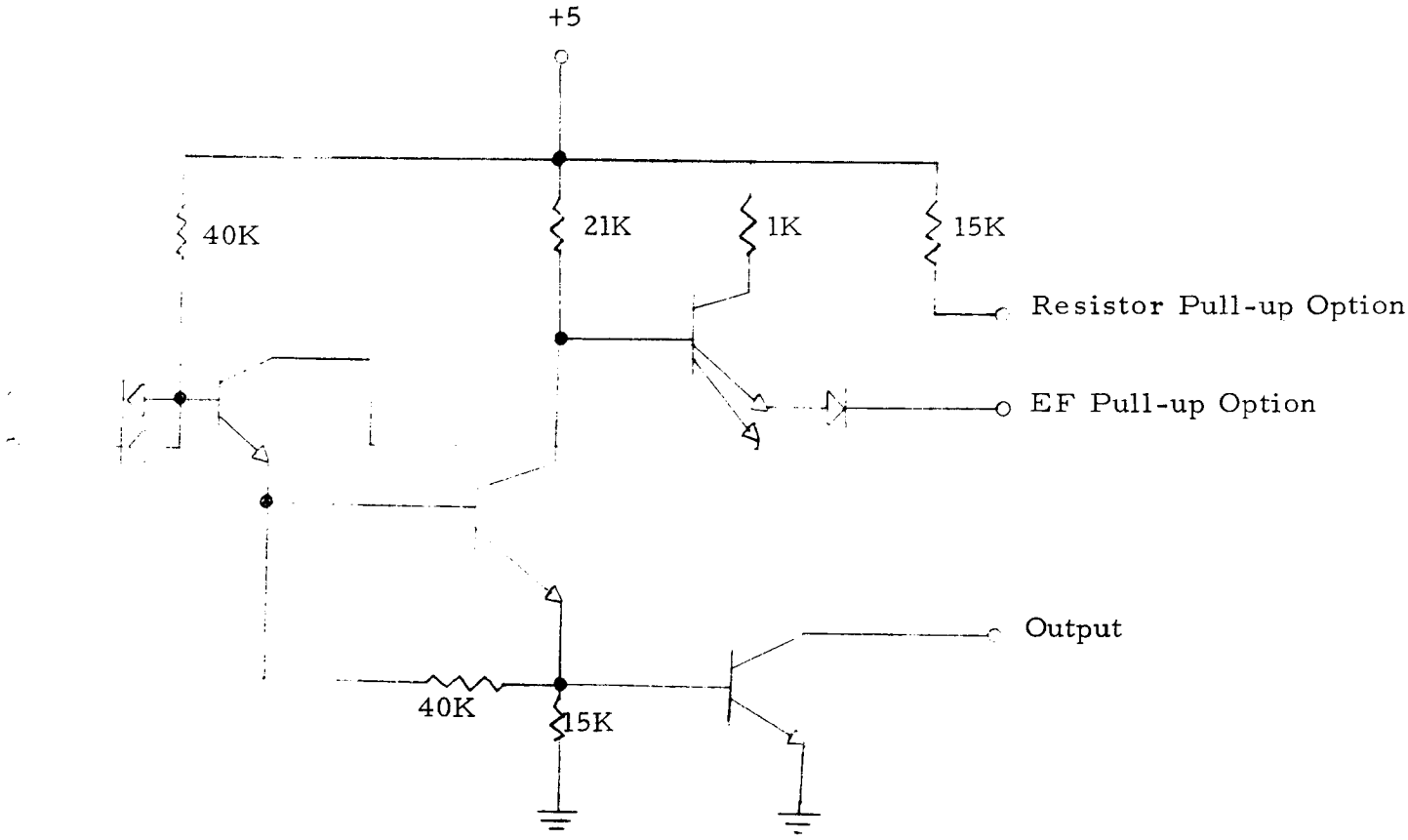


Figure 3.5-1 Basic LPDTul Circuit

the designer to choose either configuration in implementing the logic design, depending on the particular requirement. The advantage of the resistor pull-up is that, at the expense of steady-state average power requirements, connection of outputs to a common node is allowed (collector "dot"). The collector dot improves the logic efficiency of the family in that an AND function is obtained at the dot node, giving an equivalent AND-OR-INVERT function for the NAND-DOT AND that results from the dot ($\overline{AB} \cdot \overline{CD} = \overline{AB + CD}$). Thus two functions are obtained at no cost in extra logic circuits. The emitter follower output provided by the Phoenix connection precludes the use of the collector dot when the emitter follower is used, but the average power input requirement is improved and capacitive load-driving capability is provided. The power improvement is obtained because the emitter follower requires power only during the positive signal transition when driving either AND loads or capacitive loads. The steady-state UP level power requirement is very low, being determined essentially by leakage currents.

3.5.2 Input-Output Circuits

The input-output circuits required for the S^3 include a multiplexor for analog and digital inputs, analog-to-digital converter circuits, and an output driver. Each of these requirements is discussed in the following paragraphs.

3.5.2.1 Multiplexor

The multiplex function is provided by driving a MOSFET switch with logic circuits in conjunction with an interfacing circuit to provide level-translation margins at the MOSFET drive point.

The MOSFET switch has the advantage that the switch is effectively isolated from the drive circuit and the "on" switch exhibits zero offset. From this aspect, the MOSFET is an ideal multiplex element. The particular device used in this system is the General Instruments MEM 2009. The design specifications for this circuit are given in table 3.5-2.

3.5.2.2 Analog-to-Digital Converter

The analog-to-digital converter (A/D) is constructed using the General Instruments MEM 5014 multifunction LSI device, the Fairchild uA-709 integrated linear differential amplifier, and the Anastrohm LN10-01-1-100 10-bit binary ladder network. The interconnection diagram of the A/D is given in Figure 3.5-2.

The heart of the A/D is the MEM 5014. This device was developed by IBM in conjunction with the General Instruments Company for use in this type of application.

The development phase of the device has been completed and all of the original design goals have been achieved. The program is currently in the production phase with devices being advertised by the vendor. A sample order of devices produced to IBM specifications is currently on engineering environmental and life testing. Considerable testing at both the device and system level has been performed to date with satisfactory results over an operating temperature range of -55 to +85°C.

The engineering feasibility of Large Scale Integration has been established. The development program is presently moving into the qualification of a "hi-rel" device. Tests conducted to date support, with a high degree of confidence, that the device can be successfully qualified for space applications.

TABLE 3.5-2

MEM - 2009 SPECIFICATIONS

Package	- TO - 817 14-Pin Flatpac
Configuration	- 6 Analog FET gates, inputs separate, Outputs Connected to a common mode
Input Requirements	- Cutoff Voltage - VGE - -2.5 V. Min. "ON" Voltage - VGS - 20V.
Output Characteristic	- "ON" Resistance 400 max.
Maximum Allowable	- Dissipation @ 25°C ambient - 150 MW

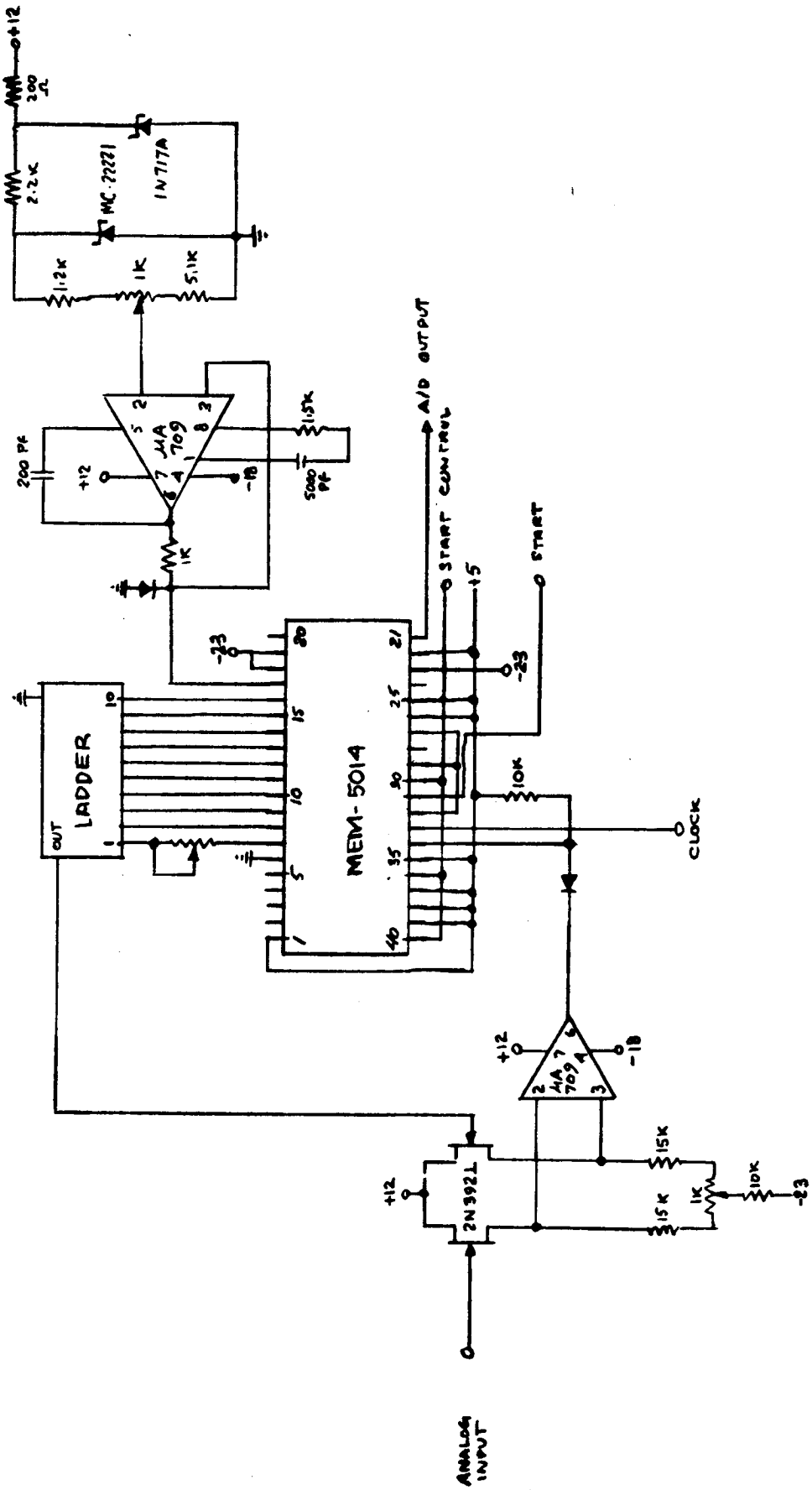


Figure 3.5-2 . A/D Converter Interconnection

The chip which has been developed for use in the A/D converter (designated Mem 5014) contains over 350 MOS transistors connected to form the basic circuitry for successive approximation conversion. The circuit contains the timing, logic and ladder switches on a single monolythic silicon chip (105x65 mils). A schematic diagram of the circuitry contained on the chip is shown in Figure 3.5-3.

In addition to the circuit chip, a ladder network, reference voltage and comparator circuit must be added to complete the A/D converter. A block diagram of the converter is shown in Figure 3.5-4 with all of the circuitry contained on the chip shown within the dotted lines.

The MEM 5014 device is implemented from P-type enhancement mode MOS transistors. Of the more than 350 devices on the chip, 20 transistors are large geometry devices which are used as the ladder switches. The ladder switches require a considerable portion of the chip surface compared to the digital circuit devices. The digital circuitry on the chip is composed of shift registers, logic gates, and timing circuits which are required to implement the A/D converter timing, logic, and holding circuits. The chip is packaged in a 40 pin in-line rectangular package. The chip itself contains 36 input-output pads. A number of the input-output lines are required for control in order to make the chip more versatile in its applications.

Operational Test Results

The accuracy of an A/D is determined primarily by the accuracies of the ladder network and the comparator circuit. Since the ladder switch "on" resistance must be included in the ladder network, the variation of this additional resistance will be the source of error

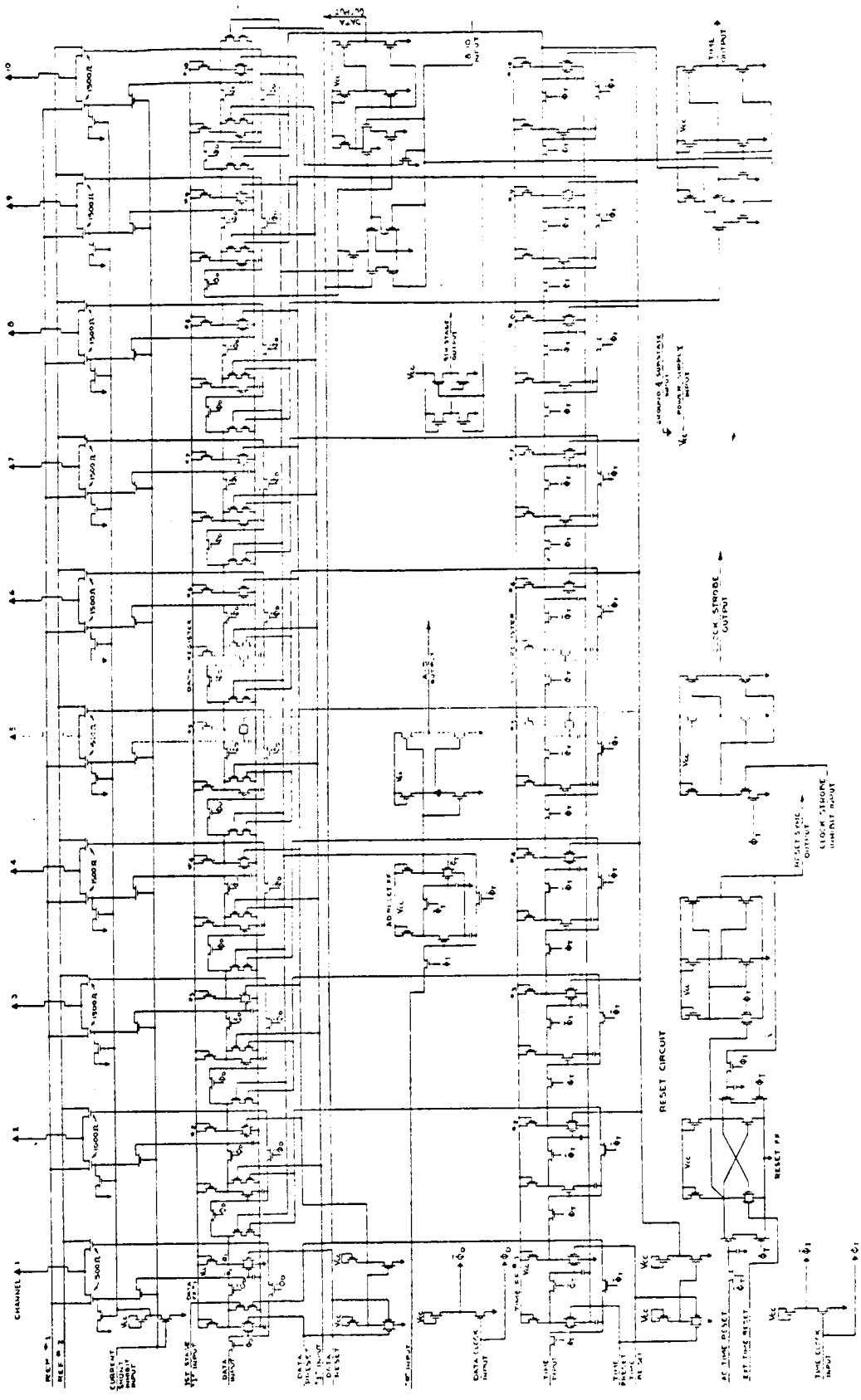


Figure 3.5-3 A/D Converter Chip

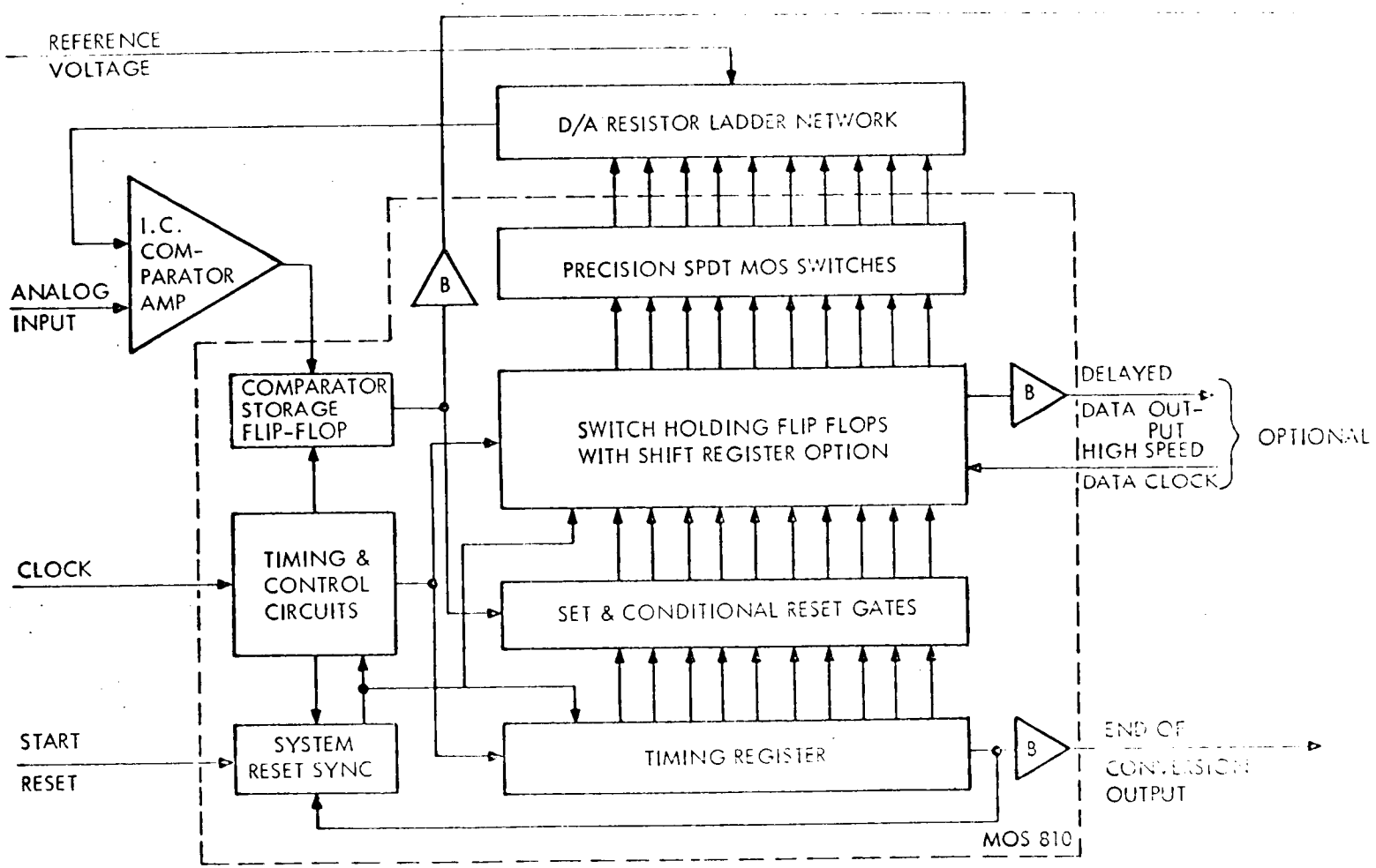


Figure 3.5-4 Block Diagram MOS A/D Converter Dotted Line Encircles Chip Circuits. 8, 9, or 10 Bit Operation Selectable

introduced into the system by the chip. There are three contributing factors to this error: variation of switch resistance with temperature, variation of switch resistance with bias voltage, which is a function of power supply voltage, and hysterersis errors caused by unequal resistance of switch pairs.

Test results on several devices show that the resistance variation as a function of these parameters is within the accuracy requirements of a 10 bit A/D converter over a -20 to +85°C temperature range when using a 50K ohm ladder network.

Thus, on the basis of these results, it can be seen that by using a fairly large value precision ladder network (50K ohm ladder with 2R = 99K ohm), a 0.05% basic accuracy over a reasonable temperature range can be achieved. In systems tests this accuracy was consistently observed using a 50K ohm ladder network and a comparator consisting of an IC operational amplifier preceded by a matched FET differential pair. This was achieved by trimming the ladder leg resistance of the most significant bit.

The speed of the system is determined by the time required for the set and conditional reset sequence. Approximately 2us are required for complete switch settling with an additional 1 us delay occurring during the system reset cycle. Since the high impedance ladder networks and comparators will have a 3 to 4 us settling time, the digital logic will operate sufficiently fast to provide a 150 kpps output bit rate.

The basic advantage in the utilization of this device for A/D conversion is that the device was designed specifically for this application and consequently contains a fairly large number of circuits connected in an optimum fashion for this requirement. Figure 3.5-5 shows a comparison

Device	Function To Be Performed	Number of Pkgs. Reqd.	External Interconnects	Power
MEM 5014	A/D Conversion (8-bit) successive approximation) less ladder network, comparator, reference supply.	1	36	120 mw
TTL Devices Series 54	Same as above	14 IC's plus transistors for switches (16)	244	1060 mw plus transistor drive power
RTL Series 51 (Low Power)	Same as above	25 IC's plus transistor switches (16)	298	150 mw plus transistor drive power
Standard MOS Devices	Same as above	21 includes MOS switches	174	250 mw

Figure 3.5-5 A/D Converter Configurations

between competing A/D converter designs. This figure points out the significant reduction in required components and interconnections brought about by the use of the MEM 5014 device.

Approximately 50 MEM 5014 devices have been packaged and tested both at the device level and in an operating A/D system. Each device is tested for the following characteristics:

1. Breakdown voltage
2. Leakage currents
3. Operating voltage range
4. Power vs. supply voltage
5. Operating speed
6. Switch on resistance.

The devices have been tested for operation over a temperature range of -40°C to +85°C. Performance and basic characteristics are satisfactory over this temperature range. Tests at room temperature after temperature cycling shows no significant change in characteristics following temperature cycling.

The MEM 5014 device tests conducted in the IBM laboratory to date have resulted in over 5000 device hours under typical room ambient operation conditions with no failures or significant changes in characteristics. In addition, approximately 1000 device hours of operation at +85°C has resulted in no change in characteristics or failures. More complete environmental and life testing data will be obtained from the first 50 production devices during the next two months.

In addition to tests on the basic chip, a complete PCM prototype system has been fabricated and tested using the MOS chip as the basic building block. This system uses the MOS chip for analog multiplexing format generation and A/D conversions. The system has been fabricated and tested with completely satisfactory results. The system accuracy

is within ± 1/2 LSB over the temperature range of -20 to +85°C.

3.5.2.3 Output Driver

In those output applications requiring power driving capability, as when it is necessary to drive capacitive cable loads, the DTL power gate using the emitter follower output connection is specified.

3.5.2.4 Interface Circuits

Two interface circuits are required, one for LPDTuL to MOSFET and one for MOSFET to LPDTuL.

LPDTuL - MOSFET Interface Circuit

The LPDTuL - MOSFET Interface Circuit is required to convert the LPDTuL down level output to a MOSFET input level equal to or more negative than -12VDC. Similarly it must convert the LPDTuL up level output to a MOSFET input level equal to or greater than +3VDC. The circuit must provide proper conversion over a -10 to +65°C temperature range with minimum power dissipation.

The circuit shown in Figure 3.5-6 is intended for this application. The description of circuit operation is as follows. When the LPDTuL output is a down level the emitter-base junction will be back biased by 3V worst case and the transistor will be turned off. The voltage at the collector will be $V_c = V_{DD} - I_L R_I$ where I_L is the sum of the leakage currents from the base-collector junction and the drain-gate junction. When the LPDTuL is an up level the emitter-base junction is forward-biased and the transistor is saturated. The collector will be clamped at approximately +3.5 VDC.

Two parts are required. The transistor is a 2N2907A and the resistor is 604K, ±5% EOL, 1/8W. Total power dissipation with the LPDTuL output at an up level is 1.5 mW per interface circuit. Power dissipation due to leakage currents when the LPDTuL output is down is neglected.

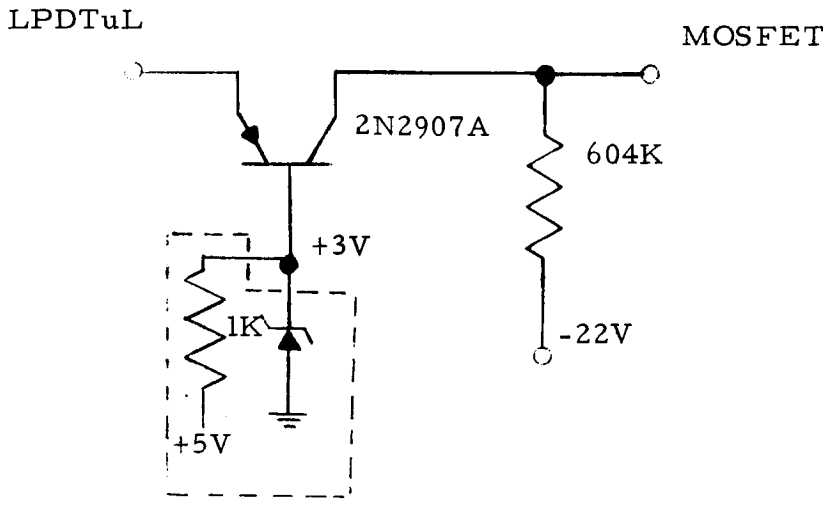


Figure 3.5-6 LPDTuL - MOSFET Interface Circuit

MOSFET - LPDT⁄L Interface Circuit

The MOSFET-LPDT⁄L interface Circuit is required to convert the MOSFET output down level to a LPDT⁄L input down level of +.5 VDC to 0.0 VDC. Similarly it must convert the MOSFET up level output to a LPDT⁄L input up level of +3 VDC to +5 VDC. The circuit must provide proper conversion over a -10°C to +65°C temperature range with minimum power dissipation.

The circuit shown inside the dashed lines in Figure 3.5-7 is intended for this application. The description of circuit operation is as follows. When the MOSFET output is an up level the PNP base emitter is zero biased and neglecting leakage currents, the LPDT⁄L input will be +5V. Figure 3.5-8A shows the equivalent circuit under these conditions. When the MOSFET output is a down level, the circuit of Figure 3.5-8B applies. The PNP is a current sink which speeds the fall of the LPDT⁄L input signal and clamps it slightly below ground. The 200K resistor limits current into the MOSFET and affects the degree of saturation.

Three parts are required. The transistor is a 2N2907A, the resistor is 200K, ±5% EOL, 1/8 W, and the diode is FD600. Total interface circuit power dissipation, neglecting leakage currents, is 0.0 mV with the MOSFET at an up level and 2.5 MW with a MOSFET down level output. This does not include the LPDT⁄L power.

3.5.3 Crystal Oscillator and Buffer Circuit

The S^3 oscillator will be the source for the Data Synchronous Clock and System Clock circuits. The DSC requires a frequency in the low megahertz (MHz) range. The highest System Clock frequency is 0.8 MHz. A 1.6 MHz oscillator was chosen to meet these requirements. An AT-cut crystal operating at 1.6 MHz maintains an accuracy of 50 ppm (.003%) over the -10°C to +65°C temperature range. The buffer is required to

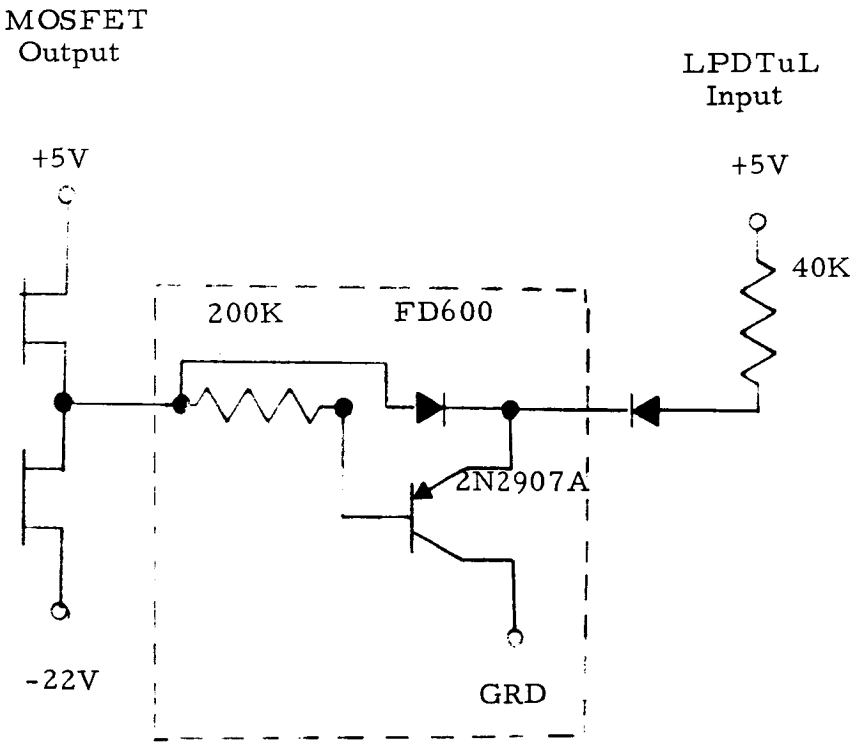


Figure 3.5-7 MOSFET - LPDTμL Interface Circuit

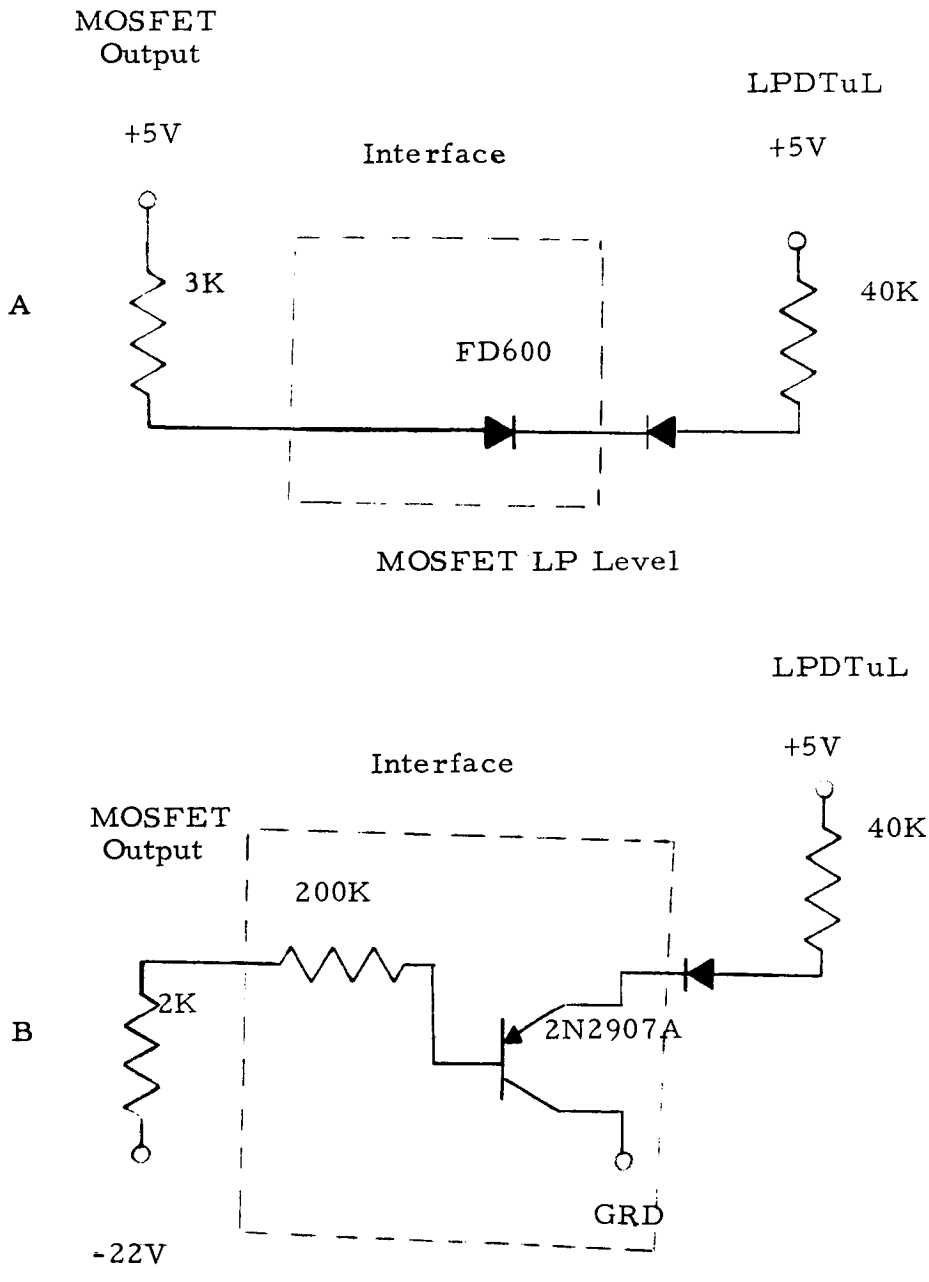


Figure 3.5-8 MOSFET Down Level

prevent overloading the oscillator, to provide level shifting of the oscillator output, and to provide drive capability.

The circuit schematic is shown in Figure 3.5-9. The components shown are typical and do not necessarily represent optimum values. They are intended only as an indication of magnitude. The table shown with the figure shows the operating conditions which influence component values. V_{out} is a 1.6 MHz square wave with pulse parameters corresponding to the standard logic module.

3.5.4 MOSFET Radiation Tolerance

It has been recognized that MOSFET devices are susceptible to failure in an ionizing radiation environment such as that encountered in the Van-Allen belts. The system requirement is that no failures should occur because of radiation for a period of one year in the worst part of the Van Allen belts. This is equivalent to a total dose of approximately 3×10^{15} electrons per square centimeter. A more realistic specification of the environment, taking into account actual orbits, should reduce the expected dosage by one or two orders of magnitude. In any event, the susceptibility of the MOSFET devices is required to be determined.

Two results of the first test (1) indicated that the insulating material exhibited about the same radiation susceptibility as silicon dioxide,

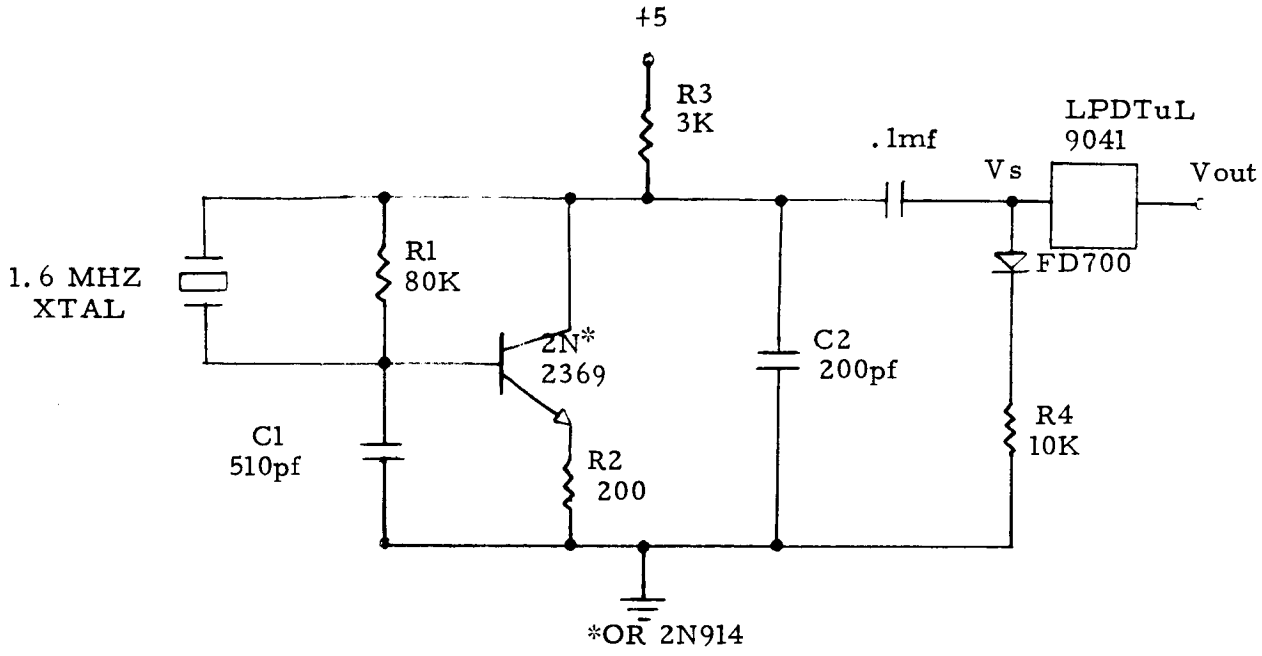


Figure 3.5-9 Oscillator-Buffer Circuit

Component	Comments	Condition Setting Upper Limit	Condition Setting Lower Limit
R1	.1W, 5%	Must provide sufficient bias for oscillation at worst case transistor beta	Must be much greater than impedence of crystal 2K
C1		Limits loop again	$C_1 + C_2$ must be much larger than crystal C_o 5 pf
C2	Most critical frequency factor	Limits loop gain	C_2 large to reduce effect of stray capacitence variation
R2	.1 watt, 5%	Limits loop gain	Stability
R3	.1 watt, 5%	Transistor bias current	Power dissipation esp in crystal
C3	AC coupling	Physical size	Proper coupling
R4	.1 watt, 5%	R4 will be chosen initially to set the average value of Vs at approximately 1 volt to get a current V out	

in fact, it was established after the test that the material was, indeed, silicon dioxide. The dosage at which failure occurred was about 5×10^{12} e/cm^2, other conditions being nominal (temperature 25°C, voltages minimal).

An investigation was made into the possibility of using Silicon Nitride as an insulator instead of Silicon Dioxide, since the literature (2) indicated that considerably better radiation tolerance could be expected from nitride-insulated devices. Although single FETs can be made using nitride, the state-of-the-art is not sufficiently developed to enable the use of nitride on LSI devices. It is estimated that about two more years of development will be required before the nitride process can be applied to LSI devices.

The only alternative left to guarantee operation under the stated conditions is to incorporate shielding. A shielding analysis (3) was performed using as a model the equivalent of a delta-pack, consisting of a multilayer board 60 mils thick on each side of a 100-mil aluminum web, plus an allowance of 3/32 inches for skin and structure. The analysis indicated that lead would be required to reduce the total radiation level by four orders of magnitude at the device. The second test was performed to gain a more accurate measure of the effects of the actual shielding materials used.

The second test (4) consisted of three parts. In the first part, a measure of the attenuation effects of the shielding was obtained. The second and third parts were to measure the failure level for two different shield configurations, one corresponding to the minimum shielding obtained by centrally locating the FET devices and the other corresponding to the minimum plus the maximum amount of shielding that could be allowed within the weight limitations of the system. The results of the test were surprisingly good. No failures were obtained using the minimum shielding, an incident dose of 10^{15} electrons per square centimeter on the outside of the shield,

an ambient temperature of $0^{o}C$, and the power voltage five percent low. The minimum shielding consisted of the equivalent of two delta-packs, since this form will be used in the system, giving two for shielding the centrally located FET's in each direction. The complete test included one MEM-5014, one MEM-5015, two MEM-3012 shift registers, and two MEM-511 FET's. The latter were included to get turn-in voltage measurements after radiation. Complete results of this test are to be contained in a separate report which will be published in the near future. Delay in receipt of some of the data has precluded incorporation of the full test results in the report.

As a result of the second test, it is tentatively concluded that no special precautions need be taken with the MOSFET devices in order to ensure their operation in the required environment other than to locate them in delta-packs such that the material of at least two delta-packs are interposed between the FET's and the outside of the spacecraft.

No analysis has yet been made to determine the reason that the good results of the second test seemed to contradict the shielding analysis. It can only be speculated at this time that either the epoxy boards accomplished more shielding of bremsstrahlung than expected, or the x-rays generated did not have the same effect on FET's as does electrons. The latter is the more acceptable explanation, but more testing beyond the scope of this contract would be required for verification.

References:

1. "Radiation Test of 810 IGFET CHIP" IBM #66-MO2-003, Space Systems Center
2. "Comparison of MOS and MNS FET transistors under electron irradiation" A. G. Stanley, pre-print of paper to be presented at the Annual Conference on Nuclear and Space Radiation Effect, Stanford University, Palo-Alto, California, 18-21 July 1966.
3. "Shielding Against 3 MEV electrons", F. C. Tietze, memorandum to Z. T. Dearden dated 10 March 1967, IBM Electronics Systems Center, Owego, N. Y.
4. "Radiation Shielding Verification Test", IBM Space Systems Center, to be published.

3.6 Interface Requirements

The interface between the S^3 data handling system and the rest of the S^3 system include analog and digital inputs, digital outputs, and power connections. The requirements for each of these are defined in the following paragraph.

Analog Inputs

Voltage	0 to 5 Volts
Source Impedance	0 to 10,000 ohms for $\pm 1/2$ LSB
Slope (w/o Sample & Hold)	ramp $\leq$ 25 millivolts/millisecond for $\pm 1/2$ LSB sinusoid $\leq$ 8 volt-cycles peak to peak $\pm 1/2$ LSB

Level Inputs

Logical "0"	0.6V max. 0V min.
Logical "1"	2.0V min. 5.25V max.
Drive Required	logical "0": -1ma. logical "1": negligible
Noise tolerance	+0.4V
Rise Time	1.0 usec. max.
Fall Time	1.0 usec. max.

Pulse Inputs

Amplitude	5 volts PTP referenced to ground
Source Impedance	200 ohms
Pulse width	1 μs min.; 100 ms max.

Logic Outputs

Logical "0"	0.2 v. max.
Logical "1"	2.4 v. min.
"0" level load	-1.0 ma max.
"1" level load	+0.1 ma max.

Power And Output

Logical "0"	0.2 V. max.
Logical "1"	2.4 V. min.
"0" Level Load	-1.0 ma max.
"1" Level Load	+1.0 ma max.
Rise Time	1.0 usec. max. 400 PF load
Fall Time	1.0 usec. max 400 PF load

Power Supplies - Current Requirements

Power requirements for the standard four delta pack arrangement, including 64 channels, 6 accumulators, 5 DOR's, and a 4K memory are shown below:

Voltage	Tolerance	Current
+12	± 2% regulation; ±2% distribution	.056 amp, 1.69 worst case peak
+5	" " " "	.865 amp.
-6	" " " "	.066 amp.
-22	" " " "	.010 amp.

3.7 Packaging

3.7.1 Introduction

The packaging design study program for the Small Standard Satellite (S^3) Data Processing System was directed toward a minimum weight, minimum height configuration which provides the environmental protection and serviceability features required for both space and ground environments, and the modular flexibility needed for efficient execution of a variety of satellite missions.

The packaging study described herin included the following areas of investigation:

a. The mechanical layout of the Data Processing System including the component layout of the individual delta pack subassemblies comprising this system.

b. A weight analysis of each delta pack assembly including alternate design approaches where applicable.

c. A basic delta pack subassembly design including methods of structure fabrication, component mounting and component and subassembly interconnection.

d. A pluggable memory core array design providing modularity and flexibility in the Data Processing System design without additional weight or volume penalty.

e. Thermal and environmental affects analysis of each delta

pack subassembly including data comparisons where applicable of different design approaches.

f. Alternate design approaches which deviate from the prescribed design ground rules as stated in section 3.7.3.

Due to the limitations of time and funding available for this study program many possible packaging configurations including the pluggable page concept (which deviates from the design ground rules) described in the S^3 proposal could not be evaluated. However, it is anticipated that the packaging configuration of the Data Processing System as described herein will be most beneficial to GSFC in the follow on phases of the S^3 program.

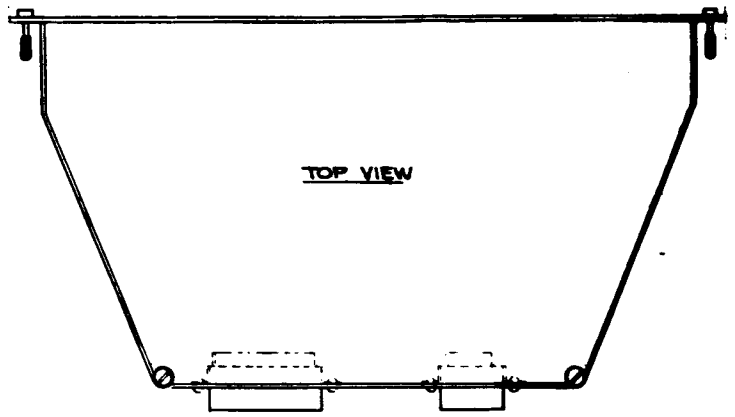
3.7.2 System Packaging

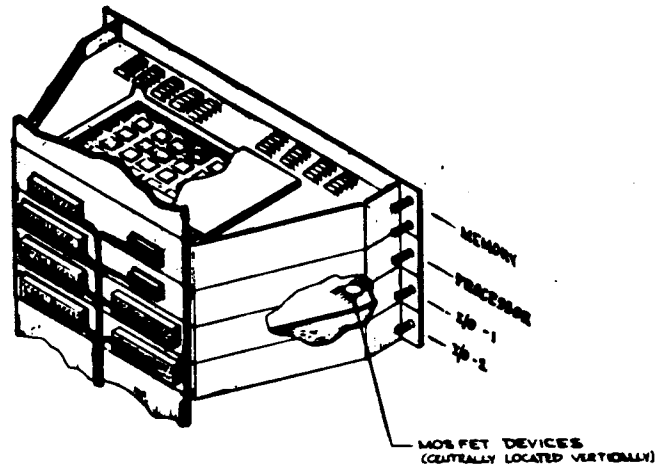
The S^3 Data Processing System is comprised of four delta pack subassemblies and arranged as shown in Figure 3.7-1. The overall weight of the system is 3.22 lbs. and the total height of the four subassemblies is 4.3 in., broken down as follows:

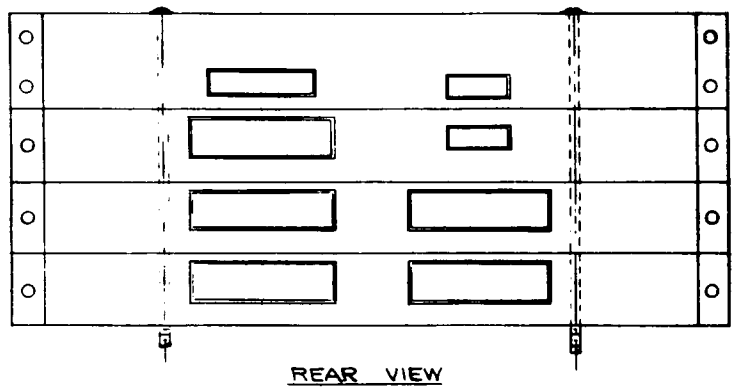
Subassembly	Wt. lbs.	Height in.
Memory	1.38	1.30
Central Control	.61	1.00
I/O #1	.67	1.00
I/O #2	.58	1.00
Total System	3.24 lbs.	4.30 in.

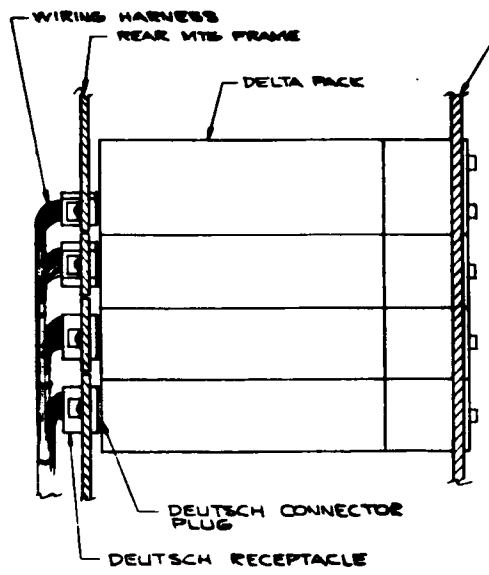
Table 3.7-1 presents a complete weight analysis of the individual components contained in each of the four delta pack subassemblies. The above figures represent weight and height saving of .76 lbs. (19%) and 70 in. (14%) when compared to the minimum data processing system design goals of 4 lbs. and 5 in. maximum height stated in the NASA S^3 Feasibility Study Report No. x-724-66-120. When compared to the 9.0 lbs. max. weight specified in the above report for a maximum data processing system the proposed system design represents a saving of 5.76 lbs. or 64%.

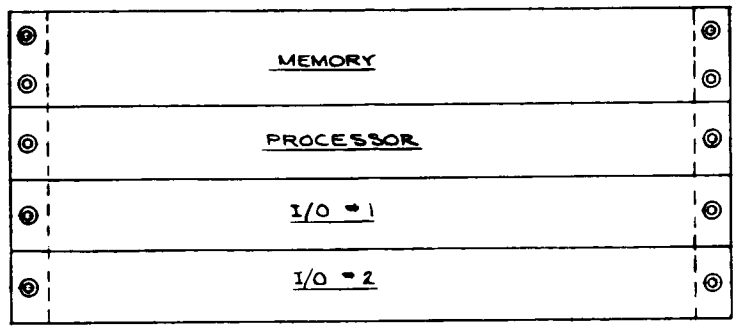
Depending upon the RFI shielding requirements for the S^3 program (see section 3.7.3), individual 1/32 in. thick magnesium covers could be added to each subassembly or a 1/32 in. magnesium shroud in the shape of a large delta pack structure could be installed in the satellite around the data processing system. This would result in a system weight increase











DATA PROCESSING SYSTEM MECHANICAL ASSEMBLY

SYSTEM WT. - 3.24 LBS
SYSTEM HT. - 4.3 IN.

FIGURE 3.7-1

M01-0071

J. SUMMERS MAR 22, 1967

SCALE: 1/1

169

of 0.6 lbs. for the cover design (2 covers per subassembly) and 0.3 lbs. for the 1/32 magnesium shroud design. With either shielding configuration the overall system weight does not exceed four lbs.

For comparison purposes a memory delta pack structure of the IMP configuration fabricated from 1/32 in. aluminum 6061 alloy (including the web) would weigh 0.22 lbs. compared to 0.31 lbs. for the proposed magnesium structure. However some degradation in heat transfer capability and mechanical rigidity as described in sections 3.7.6 and 3.7.7 would result.

Table 3.7-1

Weight Analysis of Delta Pack Subassemblies

Memory

Component	Quantity	Unit Wt. (lbs.)	Total Wt. (lbs.)
*Delta Pack Structure	1	.310	.310
Multilayer Boards (MIBs)	2	.130	.260
4K x 4 bit Memory Array	1	.067	.067
Diode Matrix Frame	1	.045	.045
Diode Matrix Insulator	1	.004	.004
Diode Matrix P.C. Board	1	.068	.068
End Caps	2	.034	.068
Spacers (array)	4	.002	.008
I/O Connector 12 pin	1	.012	.012
I/O Connector 42 pin	1	.032	.032
Array Connectors	2	.020	.040
P.C. Cables and Jumper Strip	8	.025	.200
Integrated Circuit Flat Packs	160	.0002	.030
Resistors	270	.0006	.162
Diodes	33	.0004	.013
Capacitors	18	.0006	.011
Miscellaneous	--	----	.050
Total			1.380 lbs.

Table 3.7-1 (cont.)

Central Control

Component	Quantity	Unit Wt. (lbs.)	Total Wt. (lbs.)
*Delta Pack Structure	1	.220	.220
Multilayer Boards (MIB's)	2	.130	.260
Integrated Circuit Flat Packs	276	.0002	.055
Resistors	4	.0006	.002
Capacitors	3	.0006	.002
Crystal	1	.007	.007
I/O Connector (100 pin)	1	.040	.040
I/O Connector (12 pin)	1	.010	.010
Miscellaneous	--	----	.014
Total			.610 lbs.

I/O #1

Component	Quantity	Unit Wt. (lbs.)	Total Wt. (lbs.)
*Delta Pack Structure	1	.220	.220
Multilayer Boards	2	.130	.260
Integrated Circuits Flat Packs	227	.0002	.044
Resistors	61	.0006	.037
Diodes	5	.0006	.003
Capacitors	5	.001	.005
IGFET 40-lead Device	1	.003	.003
Ladder Network	1	.004	.004
I/O Connectors (100 pin)	2	.040	.080
Miscellaneous	--	----	.014
Total			.670 lbs.

Table 3.7-1 (cont.)

I/O #2

Component	Quantity	Unit Wt. (lbs.)	Total Wt. (lbs.)
*Delta Pack Structure	1	.220	.220
Multilayer Boards (MIB's)	2	.130	.260
Integrated Circuit Flat Packs	258	.0002	.052
I/O Connector (100 pin)	1	.040	.040
Miscellaneous	--	----	.010
Total			0.582 lbs.

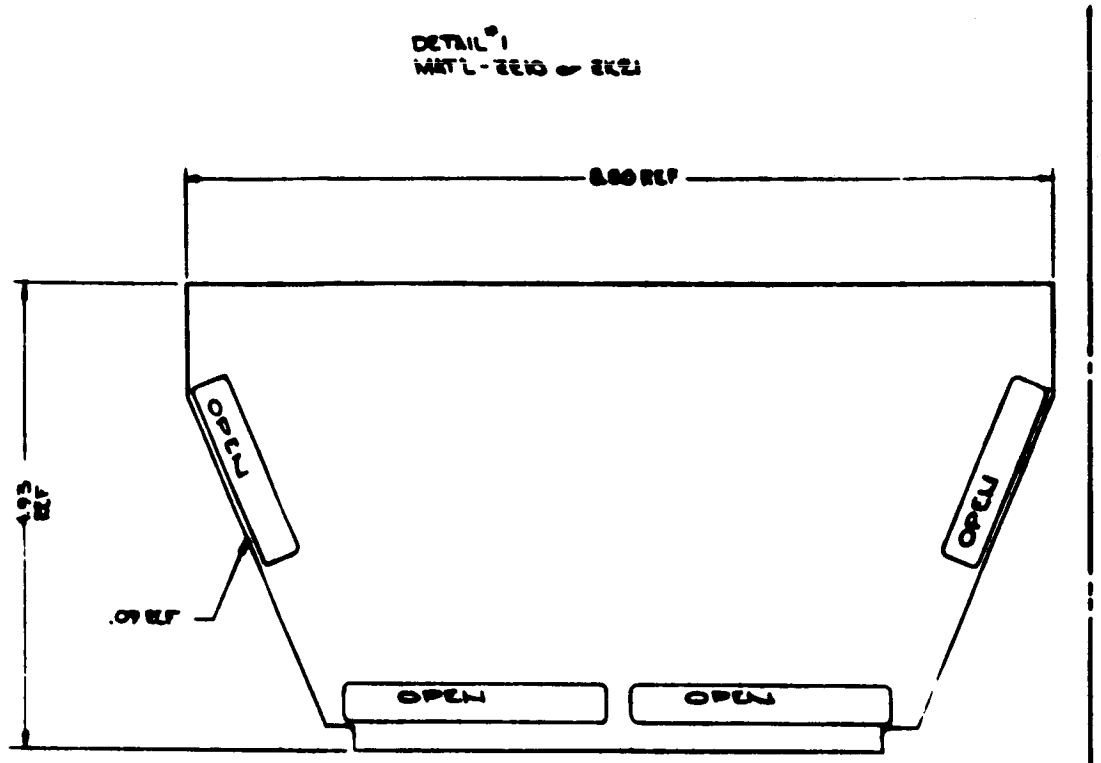
*The weight of the memory delta pack structure is based upon a structure web thickness of .093 in. in lieu of .062 for the other subassemblies. If the web thickness could be reduced to .062 in. the memory subassembly and the total system weight would be reduced by 0.09 lbs. This reduction of web thickness would be predicated upon the fact that final vibration design levels established by GSFC would be lower than the design guide levels specified in section 3.7.7.

3.7.3 Basic Package Design (S^3) Data Processing System

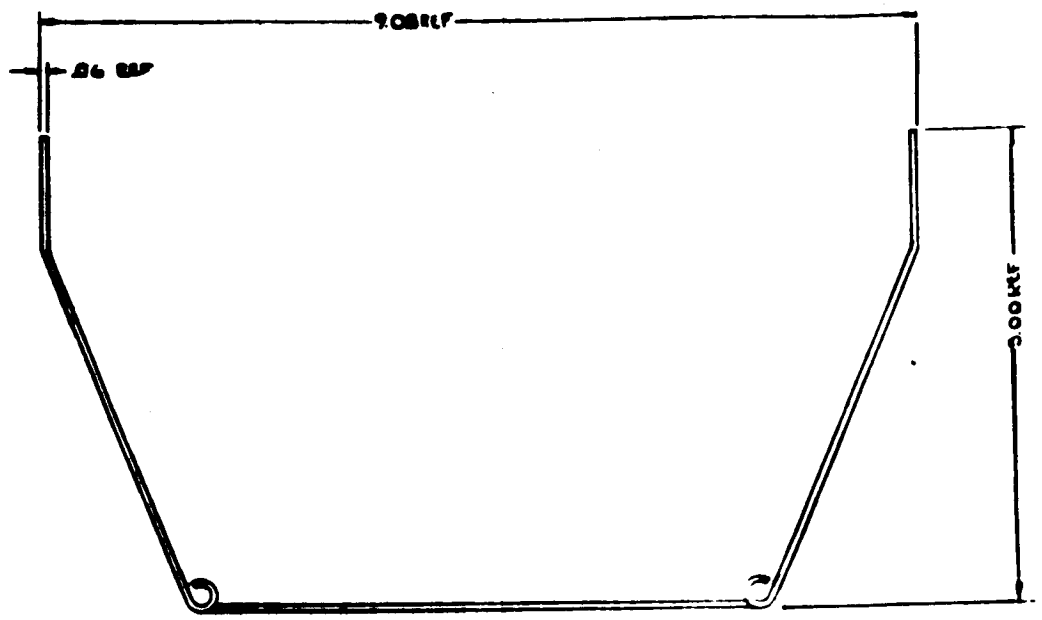
3.7.3.1 Introduction

The basic Data Processing System package configuration resulting from the S^3 mechanical design study is described in Figure 3.7-2. This configuration satisfies the requirements of minimum size, minimum weight, ease of fabrication and adaptability to the basic delta pack concept without impairing overall system reliability under the established study ground rules.

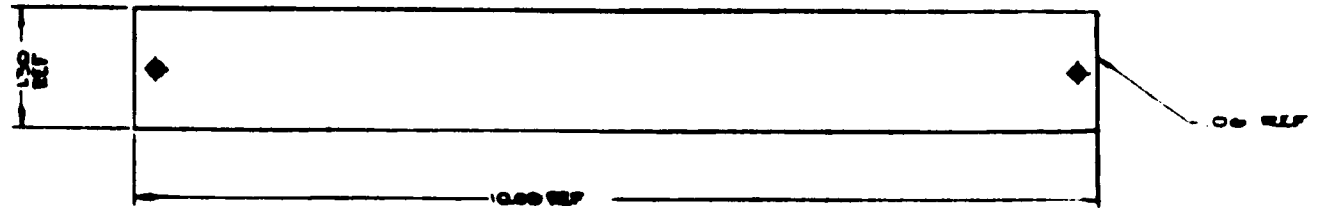
The ground rules upon which this basic package design configuration is predicated were established in meetings with GSFC and are as follows:



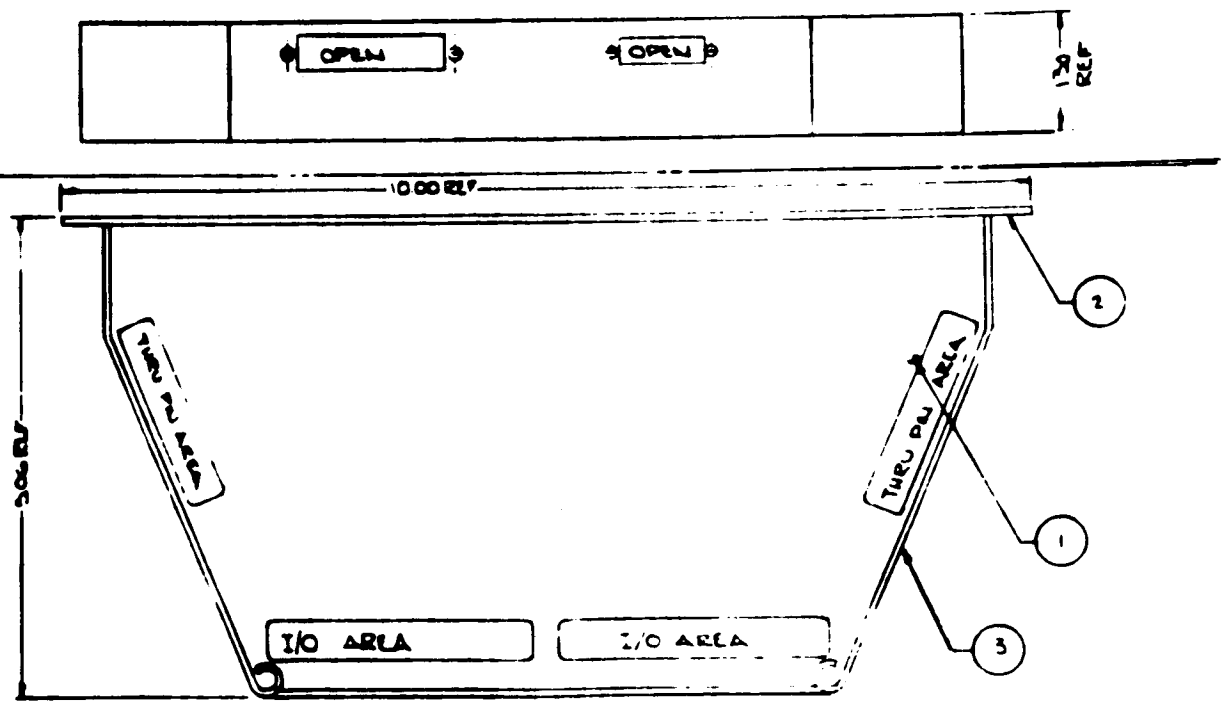
DETAIL #1
MAT'L - SEE ID or SK21



DETAIL #3
MAT'L - SEE ID or SK21



DETAIL #2
MAT'L - SEE ID or SK21



MEMORY DELTA PACK ASSEMBLY DRAWING
SCALE: 1/1

FIGURE 3.7-2

1. Delta Pack - It is desired to retain the delta pack basic form factor for the S^3 electronic subassemblies. It was suggested, however, that IBM investigate alternate techniques, other than that used on the IMP program, for delta pack fabrication while retaining the overall dimensions established by GSFC.

2. Encapsulation - The Data Processor System package design will avoid the use of encapsulants for component part support and/or component heat dissipation medium.

3. Environments - Environmental design data was not available for the S^3 satellite, only for the Scout booster. In the absence of firm environmental design data the environmental levels shown in section 3.7.6 and 3.7.7 were established by IBM and concurred with by GSFC as S^3 Data Processing System design guides.

4. Connector - It is desired by GSFC to use the Deutsch RM type connector as the delta pack electrical interface.

5. Pluggability - The delta pack subassemblies shall be individually pluggable into the satellite structure and will be secured in the front by mounting ears and in the rear by vertical tie bolts.

Packaging trade studies were performed based upon the above ground rules in which factors such as system performance, weight, volume, power, reliability, maintainability and cost were evaluated. Results of these studies and alternate packaging configurations studied are discussed in the following paragraphs. Considering the above factors, together with experience which IBM has derived from past and current space programs (Titan,

OAO, Gemini, Saturn V and MOL), the basic design configuration is the recommended packaging design for the Data Processor System in terms of conforming to the design ground rules and taking maximum advantage of past and present IBM materials, process and packaging design technology experience.

3.7.3.2 Component Packaging

The electronic component with maximum usage in the Data Processor System is the 14 lead integrated circuit flat pack. This component lends itself to a planar circuit packaging design approach using soldered or welded connections for component attachment. Obvious advantages to this design approach are component accessibility coupled with minimum volume (minimum delta pack height) and weight. IBM is presently packaging 14-lead flat packs on planar type pluggable subassemblies called pages for its 4 Pi computer family using a hot wire or blade solder reflow technique for lead attachment. This and many other 4 Pi design/process techniques have been adapted for the S^3 package design. The proposed delta pack structure consists of a trapazoidal shaped frame with an integral center support plate or web. In addition to adding mechanical integrity to the delta pack structure, the web serves as the component interconnection board mounting surface and provides a thermal conduction path between the heat dissipating electrical components and the delta pack mounting/heat transfer surfaces.

The integrated circuit flatpacks are solder-attached to etched patterns on the surface of multilayer interconnection boards (MIB's). One

MIB is bonded to each side of the delta pack structure web. An epoxy glass insulator separates each MIB from the web. The use of multilayer interconnection boards allows the flat packs to be closely spaced on the MIB surface and thus improves packaging density. Each delta pack MIB is capable of mounting approximately 142 DTuL flat packs. Discrete components such as resistors, capacitors, etc. are soldered in plated holes or to mounting pads on the board surfaces, depending upon their lead configuration. A conformal coating is applied over each board surface for component support and environmental protection. Coating material removal for component replacement is effected with a heated soldering iron.

The multilayer printed circuit boards are made of several layers of etched copper clad epoxy-glass laminates separated by pre-preg insulators which are bonded together under heat and pressure. The number of interconnection layers depend on interconnection density, number of voltages required and use of ground layers. Connections between conductor layers are made through plated holes. These boards have been in volume production at IBM since 1962 and have been successfully used in flight computers produced by IBM for the Titan II, Titan III, Gemini, Saturn I, and Saturn V programs.

Areas for feed-through pin-type interconnections between MIB's and circuit function test points are provided in the areas indicated in Figure 3.7-3. Additional areas are provided for the solder attachment of discrete wire jumpers from the delta pack Deutsch Type RM I/O connector's and the MIB's. These jumper wires will be potted in the connector back shell to provide

support under vibration environment. The RM connector is a modified version of the Deutsch MR hermetically sealed connector. Precise connector locating dimensions are not shown since they must be referenced to the mating receptacle locations on the satellite structure. The I/O connectors are shown mounted on the inner flange surface of the delta pack above the plane of the web.

This location of the Deutsch RM connector results in an overall system height penalty. Techniques examined to eliminate this penalty include mounting the connector on the center of the delta pack web. If this connector mounting location were used the overall height of the assemblies which utilize only flat mounted components (i.e., Processor subassembly) would be reduced by .25 in. However, relatively large holes would be required in the structure web and in the MIB's to provide clearance for I/O jumper wire fan out. These holes would not only impair the mechanical strength and structural rigidity of the units, but reduce the overall component mounting area by 10-20 %. This connector location was, therefore, rejected. Other techniques for reducing the system height penalty include use of alternate connector designs which violates the design ground rules.

The delta pack assemblies are secured to the satellite frame by captive screws inserted through the mounting ears of the delta pack structure. A jack screw arrangement can be provided as a mechanical aid for delta pack installation and/or removal. The delta pack stack is then secured to the satellite base plate (as described in the S^3 Feasibility Study) by mounting bolts

inserted through the two clearance holes in the rear of the individual delta pack structures. The individual delta pack assemblies may be RFI shielded by means of flat trapazoidal covers secured by mounting screws to mounting pads along the edges of the structure flange which are spaced to effect interface RFI integrity. These covers when attached to the delta pack would bear against the upper and lower edges of the delta pack flange. An alternate approach to RFI sealing would be to enclose the Data Processing System subassemblies in a thin walled shroud (in the shape of a delta pack). This would effect RFI sealing of the total Data Processor System, not the individual subassemblies and, hence, attach less weight penalty to this function. The affects of these RFI shielding configurations on system weight are described in Table 3.7-1.

In addition to the utilization and adaptation of the above 4 Pi page packaging techniques, the mechanical design study of the S^3 Data Processing System produced the following significant accomplishments:

a. A high density flat pack mounting configuration (i.e., 284 FP's per delta pack assembly including 100 through pins and functional test points and a total 200 I/O connections).

b. I/O connector/MIB interconnection scheme

c. RFI shielding arrangement

d. A delta pack subassembly utilizing the Burndy ML-98P-5 and UPC connectors presented as a design trade-off study

e. A thermal and environmental affects analysis of the basic delta pack structure as proposed by IBM and the original IMP structure modified by GSFC for the S^3 program.

f. Investigate the use of a brazed, magnesium alloy delta pack structure.

3.7.3.3 Delta Pack Structure Fabrication

As part of the Data Processing System mechanical design study, methods of delta pack structure fabrication were investigated. The basic fabrication methods included welding, casting, brazing and machining. The present IMP package structure configuration, consisting of a lightweight riveted trapezoidal frame which is dimpled around its periphery to secure printed circuit boards in "snap in" fashion was not considered for the following reasons.

1. The IMP configuration requires the use of encapsulants to provide mechanical structure integrity and heat transfer path capability. The design ground rules listed in section 3.7.3.1 prohibit the use of encapsulants

for these purposes.

2. The IMP structure configuration provides no fixed mechanical connection between the printed circuit board and the delta pack frame, barring the use of encapsulants. This makes analysis of the thermal and vibration design characteristics of each delta pack subassembly and, the sizing of the Data Processing System, in terms of weight, impossible.
3. The IMP design, assuming the use of only a single double sided printed circuit board in each delta pack would require use of several delta packs to package the number of integrated circuit flat packs employed in the Data Processing System design.

The delta pack fabrication method which initially appears to offer the advantages of mechanical strength, structural rigidity, and heat transfer capability coupled with low fabrication cost is brazing. Welding of thin walled sections (i.e., 1/16) produces severe distortion during heat treatment and poor dimensional stability. Machining the structure from a solid piece of material or casting the structure and then finish machining it is considered to be perfectly feasible but is costly in terms of machine time and/or outlay for initial pattern. Formed members riveted or mechanically fastened together with screws would not provide the continuous heat conduction

path from the heat dissipating components to the heat sink and the mechanical rigidity under vibration environment would be difficult to predict analytically.

A typical brazed structure for the Memory delta pack assembly is shown in Figure 3.7-2. The structure would consist of three formed elements; a front panel, a trapazoid frame and a central partition or web which are self-jigging for ease of assembly. These pieces could be fabricated of aluminum (i.e., Alloy 6061) or Magnesium (i.e., Alloy ZE-10 or ZK-21), however, a magnesium alloy structure would be 35% lighter than an aluminum alloy structure of identical dimensions.

Brazed structures offer the advantages of good dimensional stability, and mechanical strength equivalent to the base metal. Further, materials of different cross sectional area can be easily joined together.

IBM materials engineering is currently examining brazed magnesium alloy structures with the ultimate goal being the brazing of complete S^3 computer structures from Magnesium alloys. Thus far, excellent results have been obtained using Magnesium Alloys ZE-10 and ZK-21, with ZK-21 possessing the higher strength after brazing and heat treatment. Samples of brazed ZE-10 delta pack structures are in process at the present time and will be shown

to GSFC personnel when available.

Although many problems with brazing Magnesium Alloys still remain, the advantages in the use of a brazed Magnesium alloy structure for delta pack assemblies warrants further investigation.

3.7.3.4 Materials

Metalic Materials

Materials with a high strength/weight ratio (such as magnesium alloys ZE-10A or ZK-21) will be used for the S^3 delta pack structure design to minimize weight while ensuring structural integrity under the mechanical environments.

ZK-21 is proposed for greater strength and structural rigidity in brazed assemblies. Gemini cabin units of magnesium alloy construction have operated successfully through all flights and passed post-flight examination. Gemini computers (outside cabin) have not only withstood pre-launch environments, but every unit has successfully passed complete sea-water flooding with no structural or electrical functional degradation.

ZK-21 magnesium alloy was chosen for the delta pack structure for its brazing properties, good corrosion resistance when iridited and coated with polyurethane, high stiffness-to-weight ratio, and excellent machinability.

The properties of ZK-21 are:

Composition:	2.3% Zn .55% Zr balance Mg
Tensile strength yield:	34,000 psi
Elastic modulus:	6.5×10^6 psi
Density:	0.066 lb/in^3
Thermal conductivity:	0.28 cal/sec cm°C
Fatigue strength:	18,000 psi at 10^7 cycles

ZE-10A may also be utilized for delta pack structures because of its good thermal conductivity, high stiffness-to-weight ratio, and brazing properties. However, the mechanical properties of this alloy are not as good as ZK-21 after brazing. ZE-10A will be utilized for all sheet metal parts such as memory array end caps, spacers, etc. Properties of ZE-10A:

Composition:	1.3% Zn 0.17% rare earths balance Mg
Tensile strength yield:	22,000 psi
Elastic modulus:	6.5×10^6 psi
Density:	0.063 lb/in^3
Thermal conductivity:	0.32 cal/sec cm°C
Fatigue strength:	15,000 psi at 10^7 cycles

Non-Metallic Materials

Non-metallic materials, such as surface coatings, insulation and adhesives, have been selected for maximum environmental protection. The design criteria is stability under all temperature and vacuum conditions with a minimum outgassing characteristics.

Table 3.7-2

MATERIALS AND PROCESSES FOR S^3 FABRICATION

Part	Material	Experience	Specifications
Conformal coat	100% solid urethane (with modifications for thixotropic and room temp. cure versions) Repairable	Coated boards capable of passing cycling condensing humidity-RT to 160°F, 95% RH, for 10 days, -55 to 125°C, thermal shock, salt spray, etc. (4 Pi)	IBM Spec. 6032919 Material IBM Spec. 6009363 Process
Integrated circuit bonding	Modification of conformal coat to provide heat transfer as required	Same as conformal coat above (4 Pi)	IBM Spec. 6032919 Material IBM Spec. 6009363 Process
Headers, connecting blocks, memory frames	Glass-filled phenolic (FM 4005)	Headers, connecting blocks memory frames (Gemini and Saturn V) (LVDC/LVDA)	IBM Spec. 6009312
Bonding (general)	Epoxy, silicone urethane, etc.	Structural and nonstructural applications (Gemini, Saturn V LVDC/LVDA, and SS)	Material-dependent
Potting	Repairable urethanes (ProSeal 794)	Connector potting (Saturn V LVDC/LVDA)	MSFC 202 IBM Spec. 6009347
PC Boards	Copper clad epoxy glass	Double-sided circuitry (OAO & SS)	IBM Spec. 6009335 Broader thickness range than MIL-P-13949
PC Boards	Copper clad epoxy glass and epoxy glass prepreg	Multilayer fabricating for approx 4 years (Titan, Gemini, and Saturn V LVDC/LVDA)	IBM Spec. 6009335 IBM Spec. 6009311

Table 3.7-2 (Con't)

MATERIALS AND PROCESSES FOR S^3 FABRICATION

Part	Material	Experience	Specifications
Solder	Sn62 instead of 60/40	ASQ38, Gemini (higher creep at elevated temp. higher tensile and slower inter-metallic formation rate)	QQ-S-571

3.7.4 S^3 Memory Unit Packaging

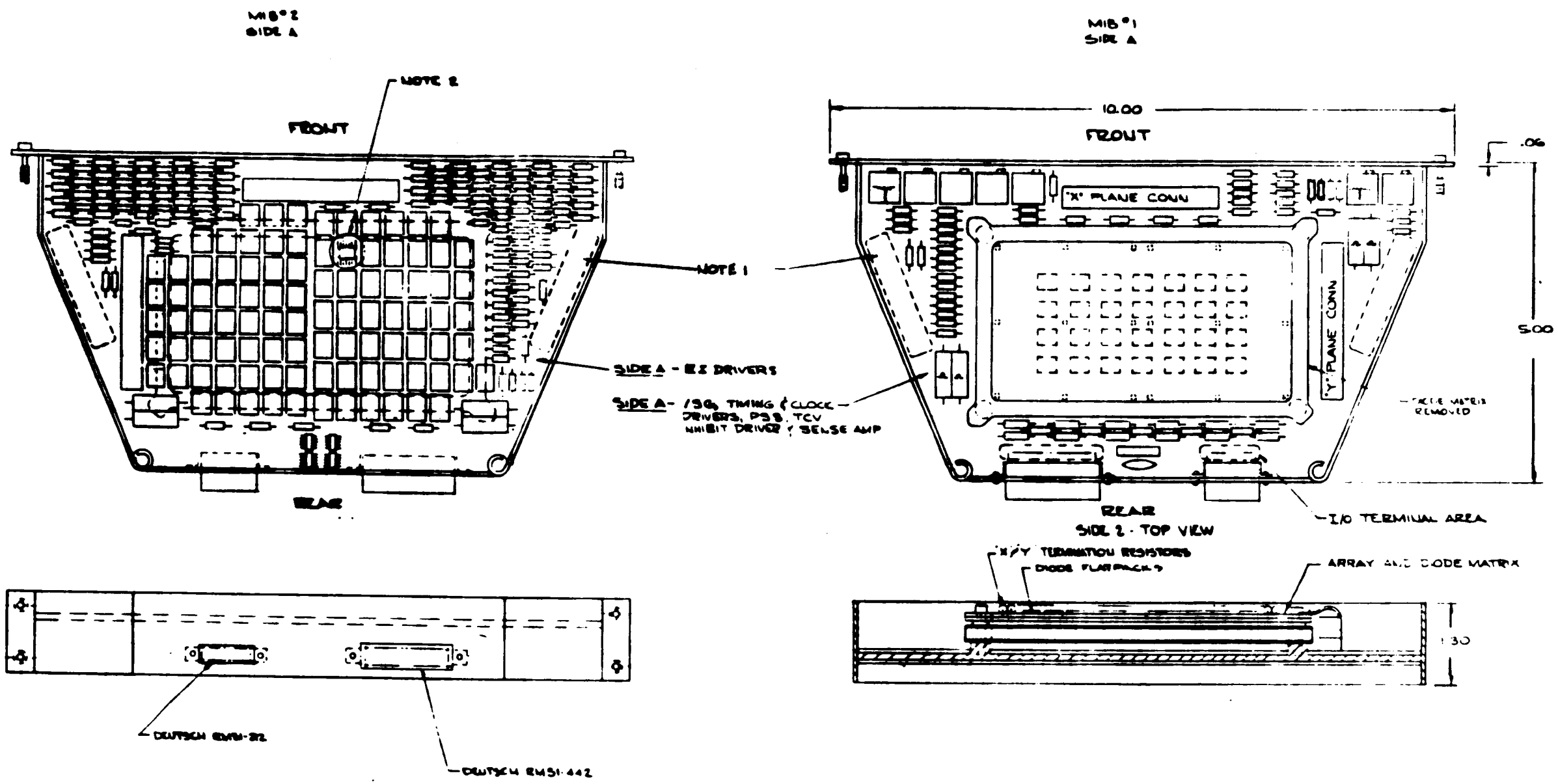
3.7.4.1 Introduction

The memory unit for the S^3 Data Processing System is described is Figure 3.7-3. The memory circuitry together with the pluggable core array is mounted on two multilayer interconnection boards (MIB's) which are bonded to a single delta pack structure. Details of the delta pack structure and MIB construction are provided in section 3.3.3. The memory unit is 1.3 in. in height and weighs 1.38 lbs. A weight analysis of this unit is covered in Table 3.7-1.

3.7.4.2 Design Features

The memory unit, as configured, combines the essential spacecraft packaging design features of minimum size and weight, high component density, serviceability, modularity and environmental integrity without impairing system performance and reliability. The integrated circuit and discrete components used in the memory unit design are selected from qualified parts used in IBM's Saturn V computer and 4 Pi computer family. The pluggable core array incorporates portions of the Saturn V, Apollo (back-up computer design study) and 4 Pi memory designs plus proven components and fabrication techniques wherever possible to optimize the memory system reliability and modularity, and to maximize core packaging density.

The memory unit as described in Figure 3.7-3 represents the maximum capacity requirement configuration for the S^3 Data Processing System, namely, the 4 K word x 4 bit system. The integrated circuit components are arranged on each multilayer interconnection



NOTES

1. TEST POINTS AND OR THRU PIN AREA
2. TYP. DUAL FLAT PACK CONFIGURATION
3. MIB ORIENTATION
   SIDE A COMPONENT ATTACHMENT SIDE
   SIDE B BONDED TO STRUCTURE (WEB)
4. I/O CONNECTOR WIRING SIMILAR TO FIGURE 3.7-2 SECTIONS A-A & B-B

M01-0063

SCALE: 1/1

MEMORY DELTA PACK - DOUBLE DENSITY MEMORY PLANE 4K x 4 BIT
WT. 1.38 LBS

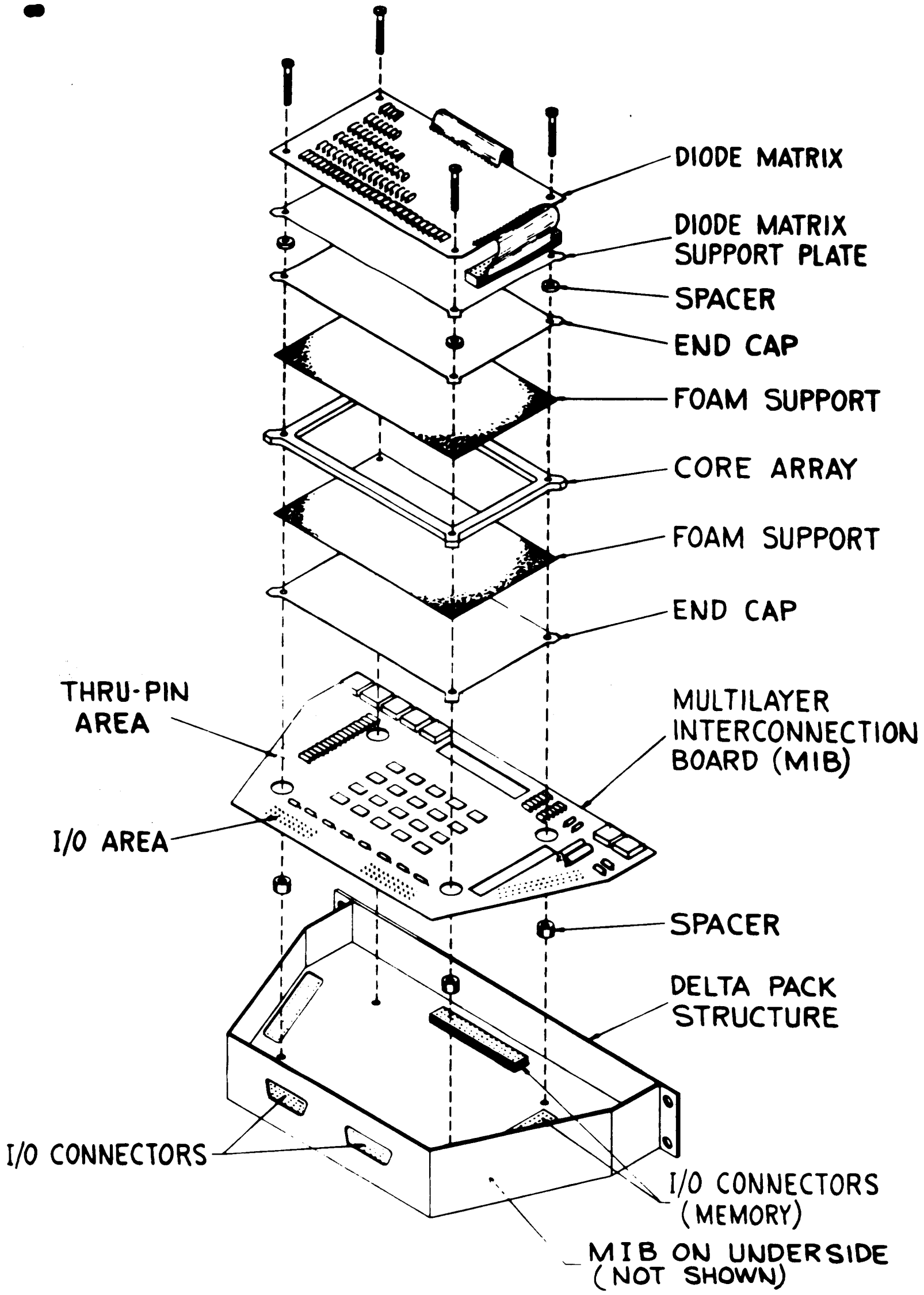
FIGURE 3.7-3

board (MIB) to optimize location of the various circuit functions and to minimize the number of through pin interconnections between MIB's. For example, the EI Driver circuits, which contain the largest number of memory components, are mounted exclusively on one of the memory subassembly MIB's while the VSG, Timing and Clock Drivers, P.S.S., TCV Inhibit Driver, Sense Amplifiers and the core array are mounted on the other memory subassembly MIB. This planar layout of the memory circuitry provides access to each component for testing or replacement. Access to those components located beneath the core array assembly is provided by loosening the four array captive mounting screws and unplugging the "X" and "Y" plane connectors. Approximately 100 test points are provided in the areas indicated in Figure 3.7-3 to permit memory subassembly probing for circuit malfunction detection. The leads of individual components may also be probed to allow further isolation of malfunctions.

Two Deutsch RM Connectors are required on the memory unit for system interconnection. A 42 pin signal connector and a 12 pin power connector will be provided in the approximate locations shown in Figure 3.7-3. Discrete wire jumpers will connect the MIB's with the I/O connectors. These connectors will be potted with Proseal 794 or equivalent (see materials section 3.7.3) for lead strain relief.

3.7.4.3 Pluggable Core Array

The modularity and flexibility of the S^3 Memory Unit is associated primarily with the core array assembly. The design of this assembly, as described in Figure 3.7-4, allows the memory unit

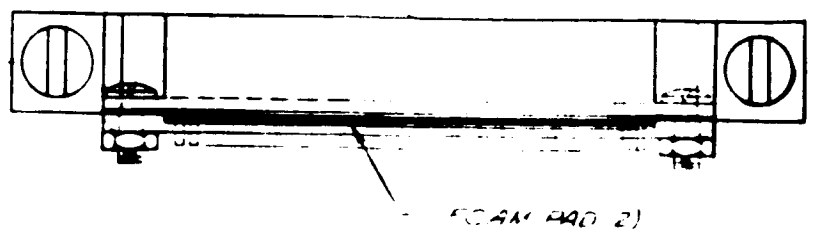


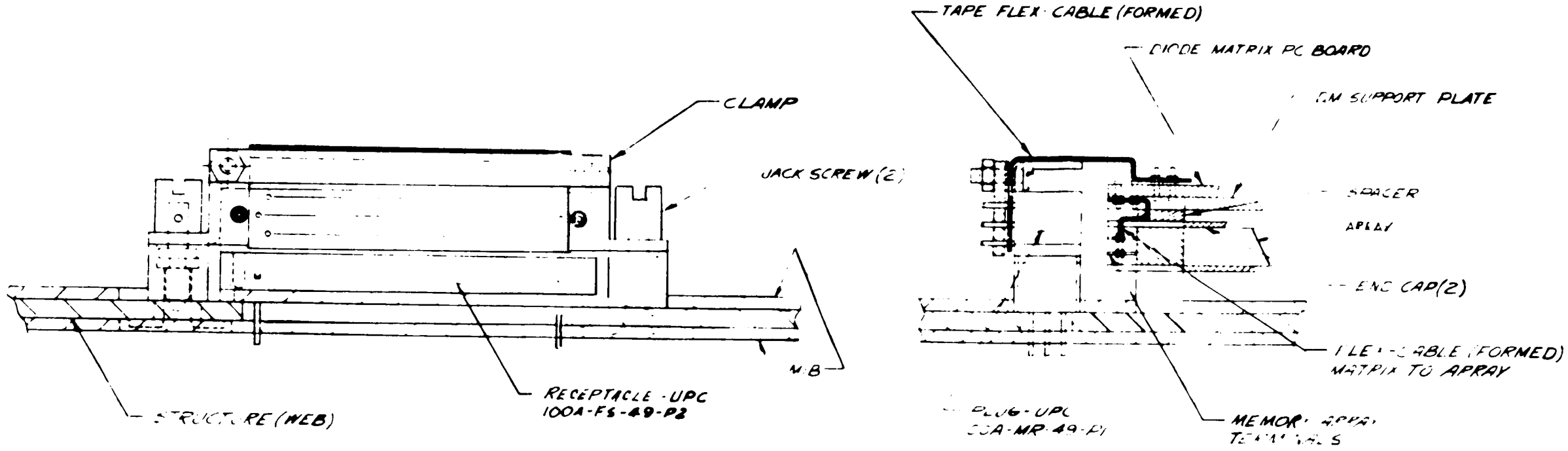
S^3 MEMORY SUBASSEMBLY

FIGURE 3.7-4

to be expanded from a 2K x 4 bit capacity to a 4K by 4 bit capacity without affecting the physical size of the memory unit and/or without changing the interconnection MIB's. The core array consists of a single core plane fabricated from a molded Fibrite 4005 frame. Terminal pins are molded into the frame to which the core matrix read, write, sense, and inhibit wires are solder attached. The core matrix area of the plane is conformally coated to provide dust and moisture protection and to preclude individual core movement on the plane wiring. To prevent excessive motion of the frame and core matrix under vibration environment foam cellular pads, bonded to Magnesium Alloy ZE-10A plates called "end caps," are mounted above and below the core plane. The memory diode matrix is attached by screws to the upper array end cap. The diode matrix components, consisting of diode flat pack and X/Y termination resistors, are mounted on a two sided printed circuit board bonded to a ZE-10 support plate. The terminal pins on the core plane are connected to the diode matrix by printed wire jumper strips which are soldered to the core array and diode matrix assemblies as shown in Figure 3.7-5. The diode matrix is connected to the delta pack MIB by two printed circuit cables which terminate in 49 pin Burndy Type UPC connectors. These connectors plug into mating receptables which are riveted to the delta pack web. The receptable pins are soldered into plated holes in the lower MIB. A jackscrew arrangement is provided (see figure 3.7-5) for mechanical aid in the insertion and removal of the array connectors.

The core plane configuration is a basic Saturn V Memory 64 x 128 plane, as modified for the Apollo Back-up Computer study program. This modification consists of the addition of 64 pins at the





MEMORY ARRAY INTERCONNECTIONS

FIGURE 3.7-5

[illegible]-0072

SCALE 3/1

narrow (Y side) ends of the frame, doubling the bit storage capacity of each plane.

Although the basic 64 x 128 Saturn core plane could be employed for the 2 K x 4 bit memory unit, the modified or "double density" core plane can be utilized for either the 2 K x 4 bit or the 4 K by 4 bit memory system by varying the number of cores wired into the matrix and by terminating the "Y" leads on alternate pins. The memory MIB's are designed to accept either core array assembly. Therefore, by the substitution of only one element of the system, i.e., the core array maximum and minimum memory system modularity is achieved without sacrificing space, weight, or serviceability. A reduction in standby power of 4% is achieved for the 2 K x 4 bit system by removing four EI drivers plus eight diode flatpacks from the assembly. These components can be left or soldered in place if expansion to a 4 K x 4 bit system is desired.

The storage element proposed for the S^3 memory unit is an IBM 13/21 lithium nickel ferrite toroidal core designated the T-80 core. The numbers refer to the inside and outside core diameter respectively in thousandths of an inch. This core is presently used in the 4 Pi computers being developed for MARK II, EA6B, and the MOL manned space program. This core was selected for S^3 because of its excellent stability over the temperature range of -55°C to +105°C, in addition to the reasons outlined in section 3.3.

3.7.4.5 Summary

In summary, the mechanical design portion of the S^3 memory study consisted of examining existing and new memory packaging techniques and conducting the necessary design trade off studies to arrive at the memory package configuration described herein. The significant accomplishments of this study are listed below:

1. Combination of the 2K x 4 bit or the 4K x 4 bit memory system into one minimum size delta pack.
2. Adapting the 4 Pi page packaging concept, utilizing planar mounted components, to the delta pack structure for maximum serviceability.
3. Proving the practicability of wiring the "double density" Apollo plane with T-80 cores, including the development of a feasible wiring matrix which utilizes different sized X/Y, sense and inhibit wires to assist in wire orientation, without impairing the electrical performance.
4. Development of a pluggable core array/diode matrix assembly for system modularity and serviceability.
5. Development of a diode matrix/core array and diode matrix/delta pack MIB interconnection technique utilizing state-of-art printed circuit cables and jumper strips.
6. Orientation of memory circuit functions to simplify the MIB wiring layout and minimize the number of through

pin interconnections between MIB's and functional test points.

7. Performance of a thermal and environmental affects analysis on the memory delta pack assembly to ensure satisfactory system performance under the assumed environmental conditions.

3.7.5 Central Control and I/O Packaging

3.7.5.1 Design Features

The Central Control and I/O Circuits are packaged in three delta pack assemblies as described in Figures 3.7-6, 3.7-7 and 3.7-8. The structures for these assemblies are fabricated as described in section 3.7.3 of this report and the circuit components, consisting primarily of integrated circuit flat packs, are soldered to etched land patterns on the surface of two multilayer printed circuit boards which are bonded to either side of the web of the delta pack structure. The details of structure fabrication, component mounting, interconnections, and environmental protection are described in other sections of this report, and will not be repeated here.

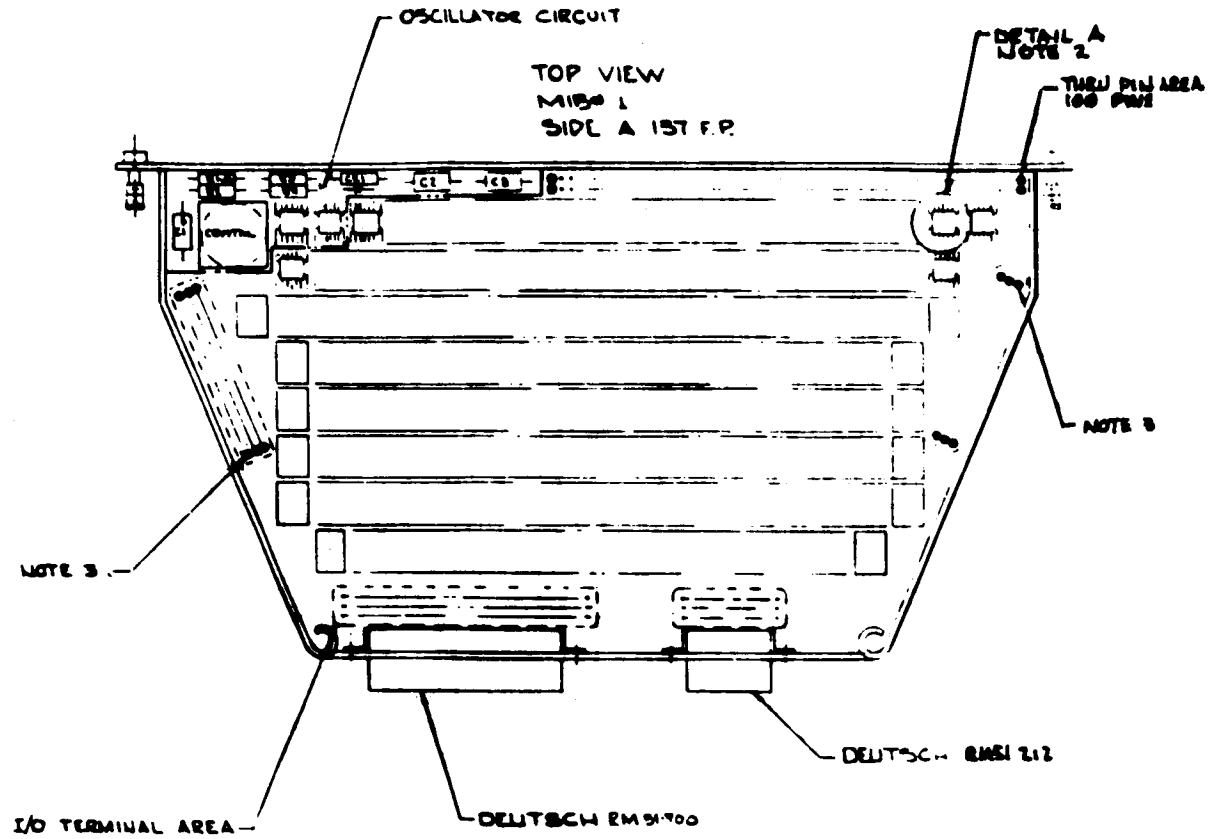
Packaging trade-off studies which were conducted, relating to the processor and I/O system design, included the determination of the maximum component density per delta pack MIB for the 14-lead DT L flatpack, the 14 or 16 lead IGFET flat pack and the multifunction IGFET 40 lead packages. These packaging studies considered providing sufficient through pin interconnections, functional test points and sufficient I/O pins for each delta pack subassembly. The results of these studies showed that approximately 142 DT L flat packs, 100-14 lead IGFET flat packs, or 30-IGFET 40 lead devices could be mounted on one delta pack MIB, while allowing 114 through pin or test point locations and space for mounting discrete components such as decoupling capacitors, load resistors etc. It was also determined that 200-I/O connections would be the maximum for any individual delta pack assembly with 150-I/O's being an average requirement (including external adaptive connections).

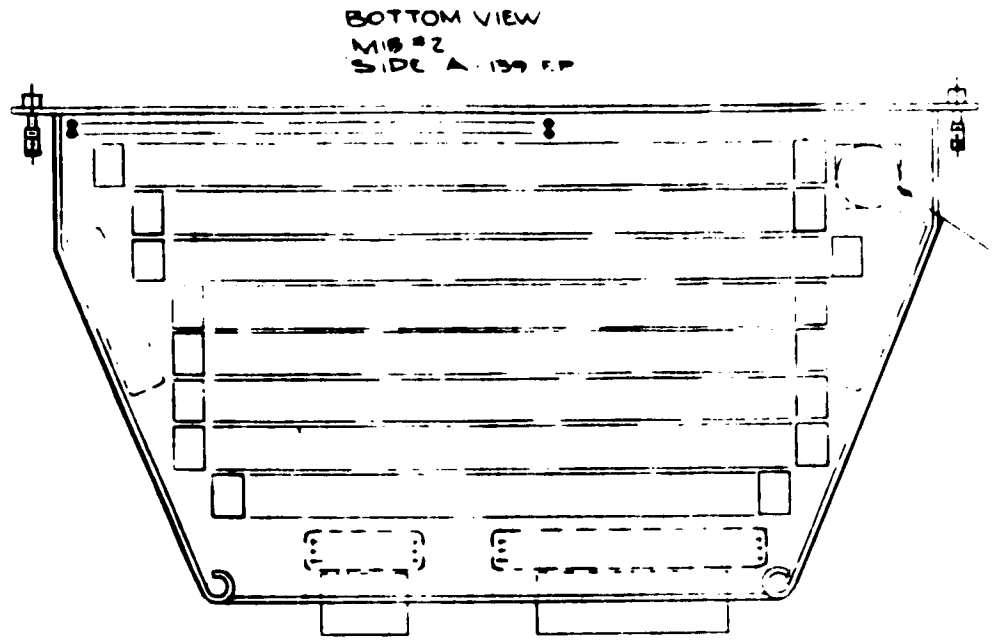
This I/O requirement can be provided by utilizing two Deutsch RM51-700 connectors each containing 100 conductor pins as shown in Figure 3.7-7.

3.7.5.2 Summary

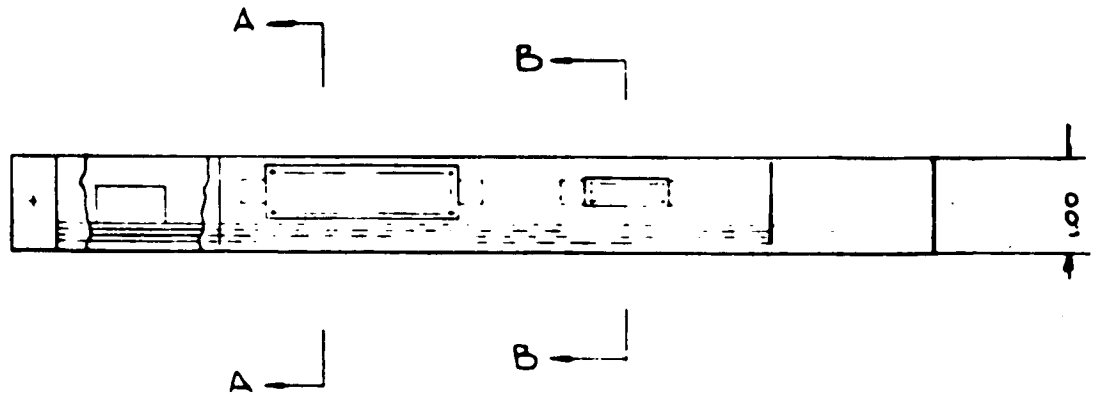
The remainder of the packaging study relating to the processor and I/O subassemblies included:

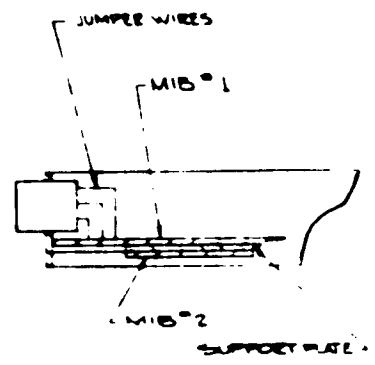
1. The layout of the Processor and I/O subassemblies based upon system component count and circuit arrangement, see Figures 3.7-6, 3.7-7 and 3.7-8.
2. Thermal and environmental effects analysis of the three individual subassemblies (described in sections 3.7.6 and 3.7.7).
3. An I/O connector trade-off study in which the Burndy ML-98P5 and UPC connectors are utilized to improve overall system reliability by simplifying the MIB to I/O connector interconnection (elimination of jumper wires) and reduce the height of individual delta pack assemblies. This study is described in section 4.7 of this report.



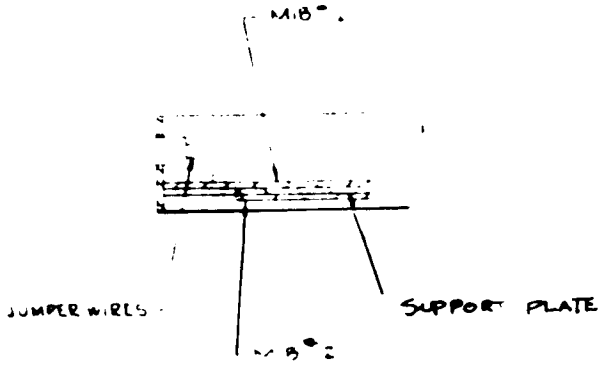


NOTES:
1. DELTA PACK CONTAINS CENTRAL CONTROL LOGIC
2. TYP DETAIL FOR PIN CONFIGURATION
3. TEST POINTS AND/OR TUNING AREA
4. MIB ORIENTATION:
   SIDE A COMPONENT SIDE
   SIDE B SOLDER SIDE
5. I/O CONNECTOR SEE SECTION A-A / B-B





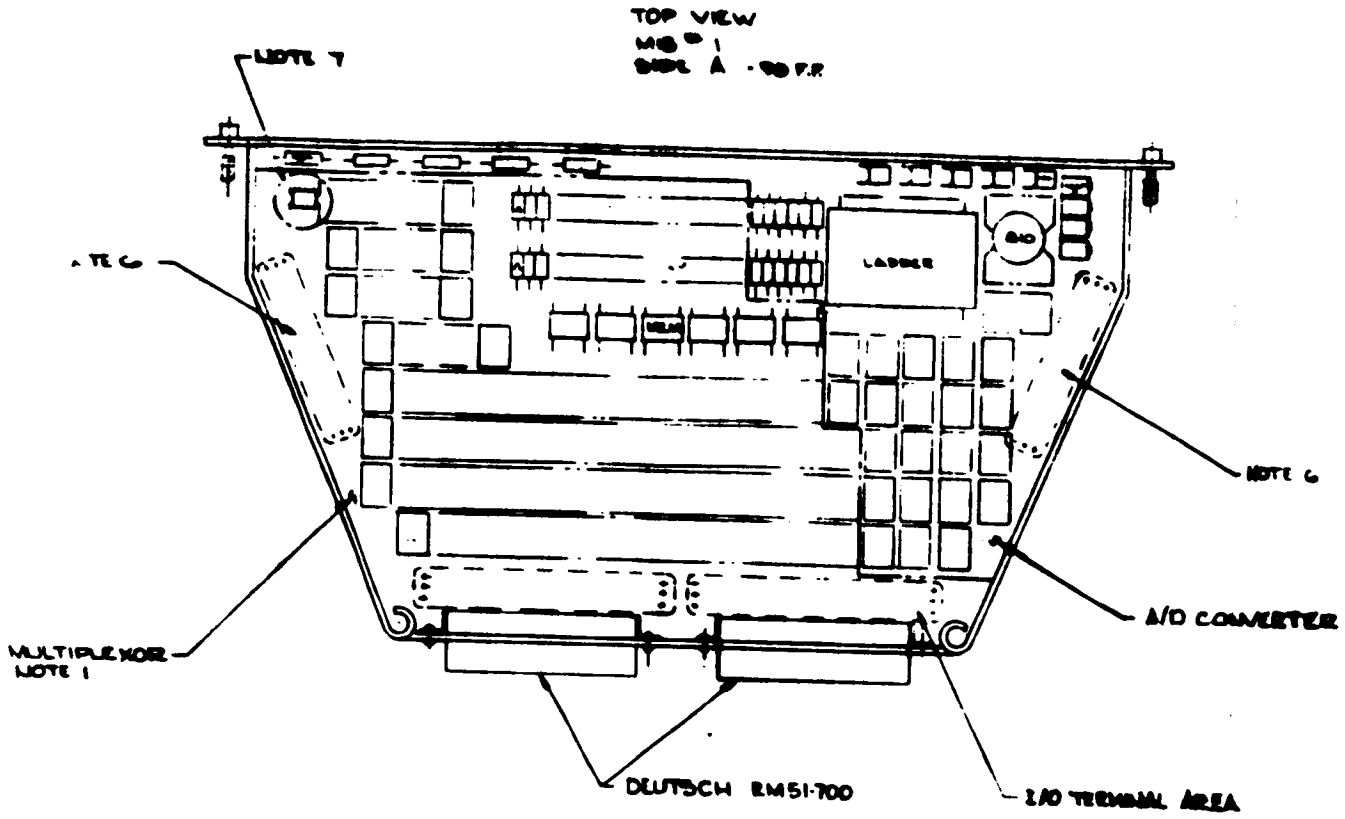
SECTION A-A
SIGNAL INPUTS

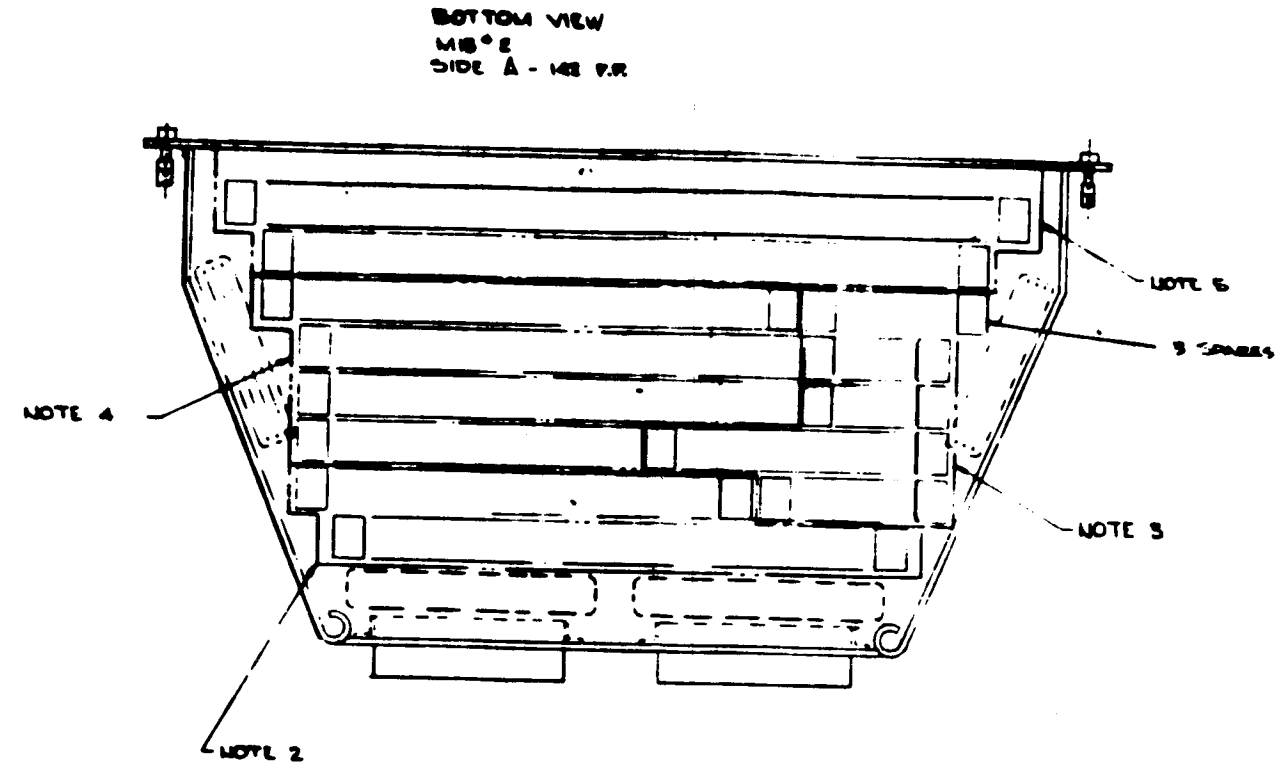


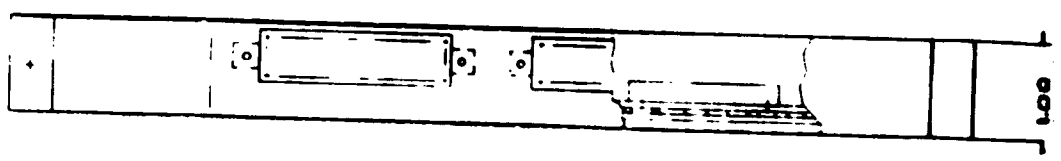
SECTION B-B
POWER INPUTS

CENTRAL CONTROL DELTA PACK
WT. .60 LBS

FIGURE 3.7-6







I/O DELTA PACK-MOD.A
WT .66 LBS

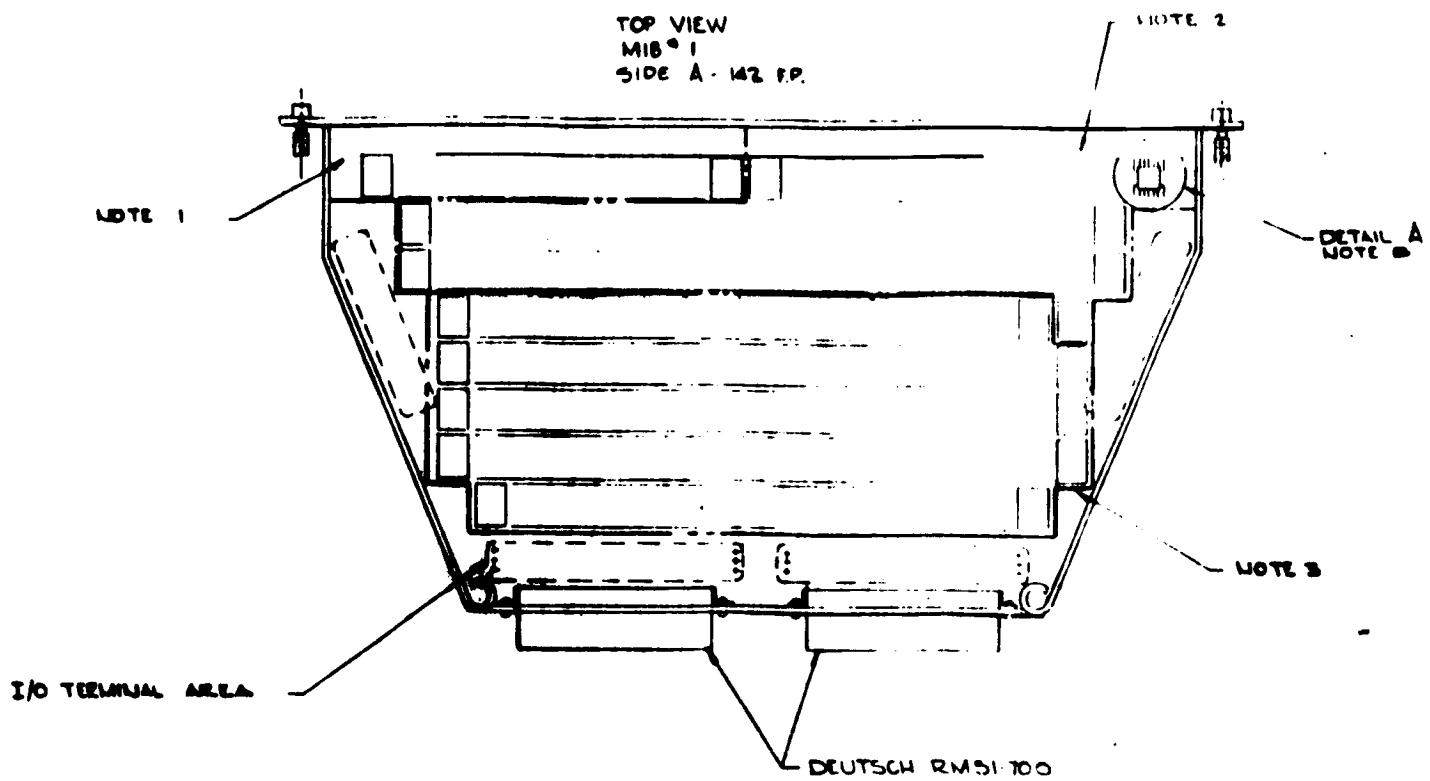
FIGURE 3.7-7

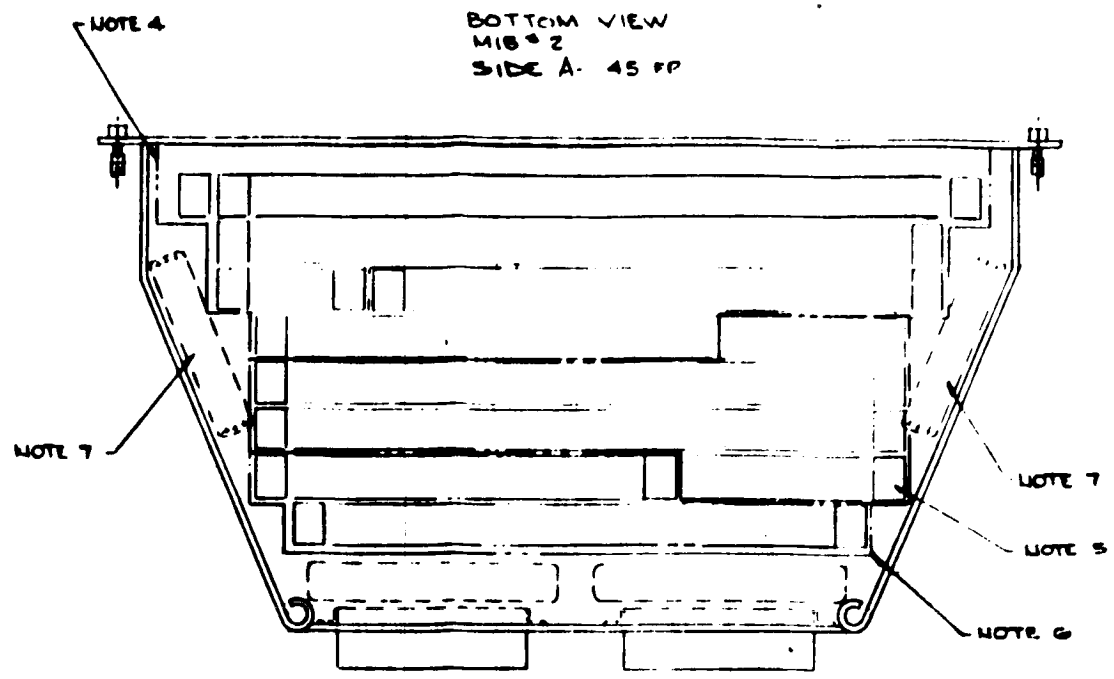
NOTES.

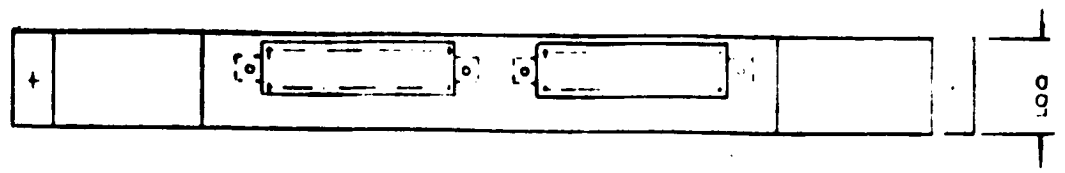
1 CONTAINS ALL OF THE ANALOG COMPONENTS PLUS 17 FLATPACKS FOR DIGITAL MUX
2 [illegible] FP REMAINDER OF DIGITAL MUX.
3 [illegible] FP COMPRESSOR
4 [illegible] FP HOUSEKEEPING
5 [illegible] FP REMAINDER OF [illegible] CONTROL LOGIC
6 TEST POINTS MOD OR THE I/O AREA
7 TYP [illegible] FLATPACK CONFIGURATION
8 MIB ORIENTATION:
SIDE A COMPONENT ATTACHMENT SIDE
SIDE B: SOLDER TO STRUCTURE INTB
9 I/C CONNECTOR WIRING SIMILAR TO FIGURE 3.7-6
SECTIONS A-A & B-B

MC1-0068

[illegible] MO1 5/17/67 SCALE: 1.1







I/O DELTA PACK - No. 2
W/T 58 LBS

FIGURE 5-1-8

NOTES:
1. 10 F.P. - WORD LENGTH STORAGE, ETC
2. 50 F.P. CONTROL COUNTERS
3. 82 F.P. ACCUMULATORS (2)
4. 71 F.P. REMAINDER OF ACCUMULATOR CIRCUITS
5. 45 FP [illegible] RITE STORAGE
6. 26 SPARE FP POSITIONS
7. TEST POINTS AND/OR THRU PIN AREA
8. TYP DTL# FLAT PACK CONFIGURATIONS
9. MIB ORIENTATION:
   SIDE A - COMPONENT ATTACHMENT SIDE
   SIDE B - BONDED TO STRUCTURE (WEB)
10. I/O CONNECTOR WIRING SIMILAR TO FIGURE 5-7-6
    SECTIONS A-A & B-B

MOI 0069

[illegible] Dr. Chk. MOI 9/11/67 SCALE: 1/1

3.7.6 Thermal Analysis

Analytical techniques have been used to predict the temperature gradients within the four delta-pack subassemblies which comprise the Data Processing System. These calculations were performed in order to ensure that the thermal design of each delta-pack subassembly is sound and that such critical components as semiconductors are maintained within specified temperature ranges under all expected environmental conditions. Verification that a sound thermal design has been achieved is demonstrated by the calculated results, which have been obtained under worst-case assumptions.

Conductive heat transfer techniques are employed to convey heat from the dissipating components to the frame of each delta pack subassembly. The resulting heat is passed from the delta-pack frame to the satellite structure at the mounting ear interface. Calculations are also included to show the effect of providing heat sinks at the I/O connector side of the delta-pack frame in addition to those at the mounting ears.

The calculations to obtain semiconductor junction temperatures are broken into four parts as follows:

1. Temperature rise from junction to flatpack case.

2. Temperature rise from flatpack case to delta-pack structure web.

3. Temperature rise from delta-pack structure web to delta-pack mounting ear or other heat sink provision.

4. Temperature rise from delta-pack mounting ear (or other point) to satellite structure.

By combining the above four temperature gradients, the temperature rise from the satellite structure to the hottest semiconductor junction is obtained.

Thus for any satellite structure temperature the hottest semiconductor temperature is known.

Typical junction to flatpack case thermal resistance is 180° C/watt for a high dissipation flatpack such as the DTuL932. In the memory and I-O #1 delta-pack subassemblies the highest dissipating flatpacks operate at 100 mw. There are three of these flatpacks in each of these subassemblies. The remaining memory flatpacks dissipate in the range of between 2 - 1/2 and 56 mw each, while the remaining I-O #1 and the I-O #2 flatpacks dissipate 4 mw or less each. In the central control delta pack subassembly, 5 flatpacks dissipate 16 mw each while all the rest dissipate 4 mw each or less. The results of the temperature rise from flatpack junction to case are given in Table 3.7-3 for each of the delta-pack subassemblies.

TABLE 3.7-3

TEMPERATURE RISE, FLATPACK JUNCTION TO CASE

	Memory	I-O #1	I-O #2	Central Control
Hottest flatpack dissipation, mw	100	100	4	16
Hottest flatpack ΔT, °C	18	18	0.7	2.9
Typical flatpack dissipation, mw	34	4	4	4
Typical flatpack ΔT, °C	6.1	0.7	0.7	0.7

Transferring the heat from the flatpack cases to the delta-pack structure web is accomplished by thermal conduction through the flatpack leads, circuit board conformal coating, and MIB. To utilize the flatpack leads exclusively to transfer all the flatpack heat to the MIB surface is not efficient because:

1. The leads, being Kovar, have a low thermal conductivity;

2. The glass insulation at the lead-case interface has a high thermal resistance.

Consequently, the relatively thick conformal coating, which is applied after all components are on the MIB, acts in parallel with the leads to conduct heat to the MIB surface. Past programs have demonstrated that a coefficient between 75 and 100° C/watt (which includes the thermal resistance of the MIB) is obtainable for the temperature rise from a flatpack case to the delta-pack structure web. These results are summarized in Table 3.7-4 for a worst-case coefficient of 100°C/watt and for a vacuum environment.

TABLE 3.7-4

TEMPERATURE RISE, FLATPACK CASE TO DELTA-PACK WEB

	Memory	I-O #1,	I-O #2	Central Control
Hottest flatpack dissipation, mw	100	100	4	16
Hottest flatpack ΔT, °C	10	10	0.4	1.6
Typical flatpack dissipation, mw	34	4	4	4
Typical flatpack ΔT, °C	3.4	0.4	0.4	0.4

From a thermal viewpoint only, the function of the metal delta-pack frame is to convey heat to the final heat sink, i.e., the satellite structure, while minimizing the temperature gradient across the delta-pack subassembly. The worst-case assumption is that all the heat generated on a delta-pack subassembly is transferred to the satellite structure at the mounting ear interface. A less severe assumption is that some of the heat from a delta-pack subassembly is transferred to the satellite structure from the connector edge of the delta-pack structure as well as from the mounting ear interface. Calculations were performed for each of these assumptions.

The memory delta-pack subassembly contains the flatpacks with the highest junction temperatures because it dissipates the most power (1.73 watts) and contains flatpacks with high dissipation (up to 100 mw), and it groups the high dissipating components, the memory array drivers, into the central region of the delta-pack subassembly (see Figure 3.7-9). The I-O #1 delta-pack assembly, though dissipating only 1.08 watts with full component complement, contains the A/D converter (500 mw dissipation), but minimizes the temperature rise within this delta-pack by spreading the A/D converter components across 1/3 of one side and placing the high dissipating components near the heat sink mounting ears.

Because of the delta pack structure geometry and the memory heat distribution characteristics, calculations for the temperature gradients within the memory delta pack structure were based on a numerical approach, i.e., the Relaxation Method. The web and flange of the delta pack structure were represented by a gridwork with a grid size of 0.8 inches, which resulted in 125 nodes. Of these nodes, 35 were located in the area of heat dissipation, while 3 nodes represented each mounting ear location. For the situation having additional heat sink capacity at the connector edge of the delta pack structure, as shown in Figure 3.7-10, 1 node represented each of the 2 additional sink areas.

The temperature of any node T_o is represented by

$$T_o = \frac{\sum_{i=1}^{4} T_i + \frac{q}{Kh}}{4} \qquad (1)$$

where

h = thickness of material

k = thermal conductivity of material

T_o = temperature of node in question

T_1, T_2, T_3, T_4 = temperatures of surrounding nodes

q = heat dissipation at node in question.

For these calculations the heat dissipating area shown in Figure A contains 35 nodes, such that the memory dissipation of 1.73 watts was evenly distributed over these nodes. As a result the values used in Equation (1) were selected as follows:

$h = \frac{3}{32}$ inch for web, $\frac{1}{16}$ inch for flange

k = 0.33 cal/sec/cm/^{o}C (80 BTU/hr/ft/^{o}F) for ZE10A

q = 0.049 watts (per node) for 35 nodes

q = 0 watts for 95 nodes.

The iterations were performed until a steady state solution of Equation (1) was reached for all nodes. This was determined by writing an error function for Equation (1) for each node as follows:

$$\Theta_o = \sum_{i=1}^{4} T_i + \frac{q}{Kh} - 4T_o \qquad (2)$$

where Θ_o is the error at the node in question. At the steady state solution, the value of Θ at each node is reduced to zero (or nearly so).

The solutions for the worst-case of heat sinks being only at the mounting ears is given in the form of isotherms in Figure 3.7-9. Only the web isotherms are given because this is where the components are placed. It is apparent that the flanges conduct a large percentage of the heat, particularly in the regions near the mounting ears. Figure 3.7-9 shows that the temperature rise from the mounting ears to the hottest spot on the structure web is 3.3° C.

A solution for the case of additional heat sink capacity at the connector edge of the delta-pack structure is shown by the isotherms of Figure 3.7-10. All conditions were similar to those shown in Figure 3.7-9 except that two additional heat sink areas, each $^{3}/_{32} \times {}^{1}/_{2}$ inch, were included. As expected, the additional heat sink capacity reduced the temperature rise from the mounting ears (or the additional sinks) to the hottest spot of the structure web. The temperature rise value for this case is 1.6° C. Note that the location of the hottest spot has shifted compared to that shown in Figure 3.7.9. Once again, a large percentage of the heat flow occurred in the flanges, particularly in the vicinity of the mounting ears and the additional heat sink areas.

Gradients for the other delta-packs were obtained using the memory results as a base. Table 3.7-5 summarizes the temperature rise data calculated for the delta-pack structures.

TABLE 3.7-5

TEMPERATURE RISE ACROSS DELTA PACK STRUCTURE

	Memory	Central Control	I-O #1	I-O #2
Dissipation, watts	1.73	1.01	1.08*	1.06*
Mounting ears as sink, ΔT^{o}, C	3.3	1.9	2.1	2.0
with additional sinks ΔT^{o}, C	1.6	0.9	1.0	1.0

*With full component complement

The remaining calculation for the conduction mode of heat transfer involves the interface between the final heat sink (satellite structure) and the delta-pack structure. This interface occurs at the mounting ears of the delta-pack structure under the worst-case assumption, and also at the additional heat sink areas at the connector edge of the delta-pack structure under the less severe assumption.

It is possible to achieve an interface coefficient up to 2000 BTU/hr/$ft^2/^oF$ under excellent conditions. IBM has achieved an average coefficient of 600 BTU/hr/$ft^2/^oF$ readily. In order to allow for the fact that the mounting ears of the delta-pack structures are rather thin (1/16th inch) and would not maintain an even pressure over the entire area of each ear, a conservative coefficient of 100 BTU hr/$ft^2/^oF$ was used. The interface equation is as follows:

$$\Delta T = \frac{Q}{AC}$$

where

ΔT = temperature rise across interface

Q = heat flow across interface

A = interface area

C = interface coefficient

Values used were 0.39 sq. in. for each ear interface area of the memory delta-pack subassembly, and 0.30 sq. in. for each ear interface area of the other subassemblies. The interface results are summarized in Table 3.7-6.

TABLE 3.7-6

DELTA-PACK INTERFACE TEMPERATURE RISE

	Memory	Central Control	I-O #1	I-O #2
Dissipation, watts	1.73	1.01**	1.08*	1.06*
ΔT across ears if ears are only sink, °C	6.1	4.6	4.9	4.4
ΔT across ears if additional sink is present, °C	3.0	2.2	2.4	2.2

*With full component complement **Some of CCU is in I-O #1.

Combining the data contained in Tables 3.7-3 to 3.7-6 gives complete temperature rise results from the satellite structure to the hottest semiconductor junction. These composite results are summarized in Table 3.7-7.

TABLE 3.7-7

TEMPERATURE RISE FROM SATELLITE STRUCTURE TO HOTTEST SEMICONDUCTOR JUNCTIONS ON DELTA-PACK SUBASSEMBLIES

	Dissipation Watts	ΔT if Ears Only Sink, °C	ΔT if Additional Sink Present, °C
Memory	1.73	37.4	32.6
Central Control	1.01**	11.0	7.6
I-O #1	1.08*	35.0	31.4
I-O #2	1.06*	7.5	4.3

*With full component complement **Some of CCU is in I-O #1.

As a design guide for the Study Program, it has been assumed that the satellite structure can have any temperature between -10°C and +30°C in vacuum conditions. The temperature range of semiconductor junctions would then result from combining the coldest (lowest dissipation) flatpack and the -10°C satellite structure limit, and from combining the hottest (highest

dissipation) flatpack and the +30°C satellite structure limit. These semiconductor junction temperature ranges are summarized in Tables 3.7-8 and 3.7-9.

TABLE 3.7-8

Temperature Range of Semiconductor Junctions For Satellite Structure Range of -10° C to +30° C if Delta-Pack Mounting Ears are Sole Sinks

	Dissipation, Watts	Coldest T_j; °C	Hottest T_j; °C
Memory	1.73	-3.8	+67.4
Central Control	1.01**	-5.3	+41.0
I-O #1	1.08*	-5.0	+65.0
I-O #2	1.06*	-5.5	+37.5

*With full component complement **Some of CCU is in I-O #1.

TABLE 3.7-9

Temperature Range of Semiconductor Junctions for Satellite Structure Range of -10° C to +30° C if Additional Heat Sinks are Present

	Dissipation, Watts	Coldest T_j; °C	Hottest T_j; °C
Memory	1.73	-6.9	+62.6
Central Control	1.01**	-7.7	+37.6
I-O #1	1.08*	-7.5	+61.4
I-O #2	1.06*	-7.7	+34.3

*With full component complement **Some of CCU is in I-O #1.

Thus it is shown that the hottest semiconductor junction is limited to 65.0°C which is well below the limit of 100°C.

General radiation from one delta-pack assembly to another will be negligible because of the small temperature differentials. There will be a favorable tendency, however, for radiation effects to minimize the temperature rise of the hotter flatpacks and thus provide a smoother thermal gradient throughout the Data Processing System than that shown in the calculated results. Conservative assumptions have been used to make the calculated data outside limits rather than nominal expected results.

Conductive heat transfer techniques have been used exclusively in the thermal analysis. If desired, it would be possible to use the exposed flange areas of the delta packs (mounting ear edge) as a thermal radiator. This would depend on the temperature ranges expected on the surface of the satellite skin which faces the delta-pack subassemblies. High skin temperatures (relative to the delta-pack temperatures) would make this undesireable. If both the delta-pack exposed flange area and the satellite skin facing the delta-pack subassemblies have high emissivity values, the required satellite skin temperature is calculated from the equation

$$T_2^4 \approx T_1^4 - \frac{Q}{E_1 A_1 \sigma}$$

where

T_1 = absolute delta-pack flange temperature

T_2 = absolute satellite inside skin temperature

E_1 = delta-pack flange emissivity

A_1 = delta-pack flange area

σ = radiation constant

Q = heat flow rate

Equation (4) is a good approximation if the emissivity of the satellite skin is high and if the effective area of the satellite skin facing the delta pack

subassemblies is large compared to the delta-pack flange area.

For the memory delta-pack subassembly dissipating 1.73 watts, having a flange area of 13.0 sq. in. (10 x 1.3 inches), a flange temperature of 36.1°C (corresponding to the ear temperature during conduction cooling), and a flange emissivity of 0.8, the required satellite skin temperature would be -13°C or lower.

For the same conditions except that the flange temperature is 68.7°C (corresponding to a semiconductor junction temperature of 100°C), the required satellite skin temperature would be +35°C or lower.

If the satellite skin temperature were high relative to the delta pack temperature, and conductive heat transfer was utilized as initially discussed, it would be necessary to limit thermal radiation from the skin to the delta-pack subassemblies by having low emissivity surfaces or by otherwise blocking this radiative thermal path.

All discussion to here has been based on a delta-pack web thickness of 3/32 inch and a flange thickness of 1/16 inch. If both these items were reduced to 1/32 inch aluminum, the temperature rise in the delta-pack frame would be increased by a factor of approximately 2.7 (assuming the aluminum has a conductivity 10% better than the ZE-10A magnesium alloy). In addition, the mounting ear stiffness would be reduced to 19% of the original value, decreasing the mounting ear interface coefficient by roughly 25%. The new results are summarized in Table 3.7-10 for the case of the mounting ears being the sole heat sink.

TABLE 3.7-10

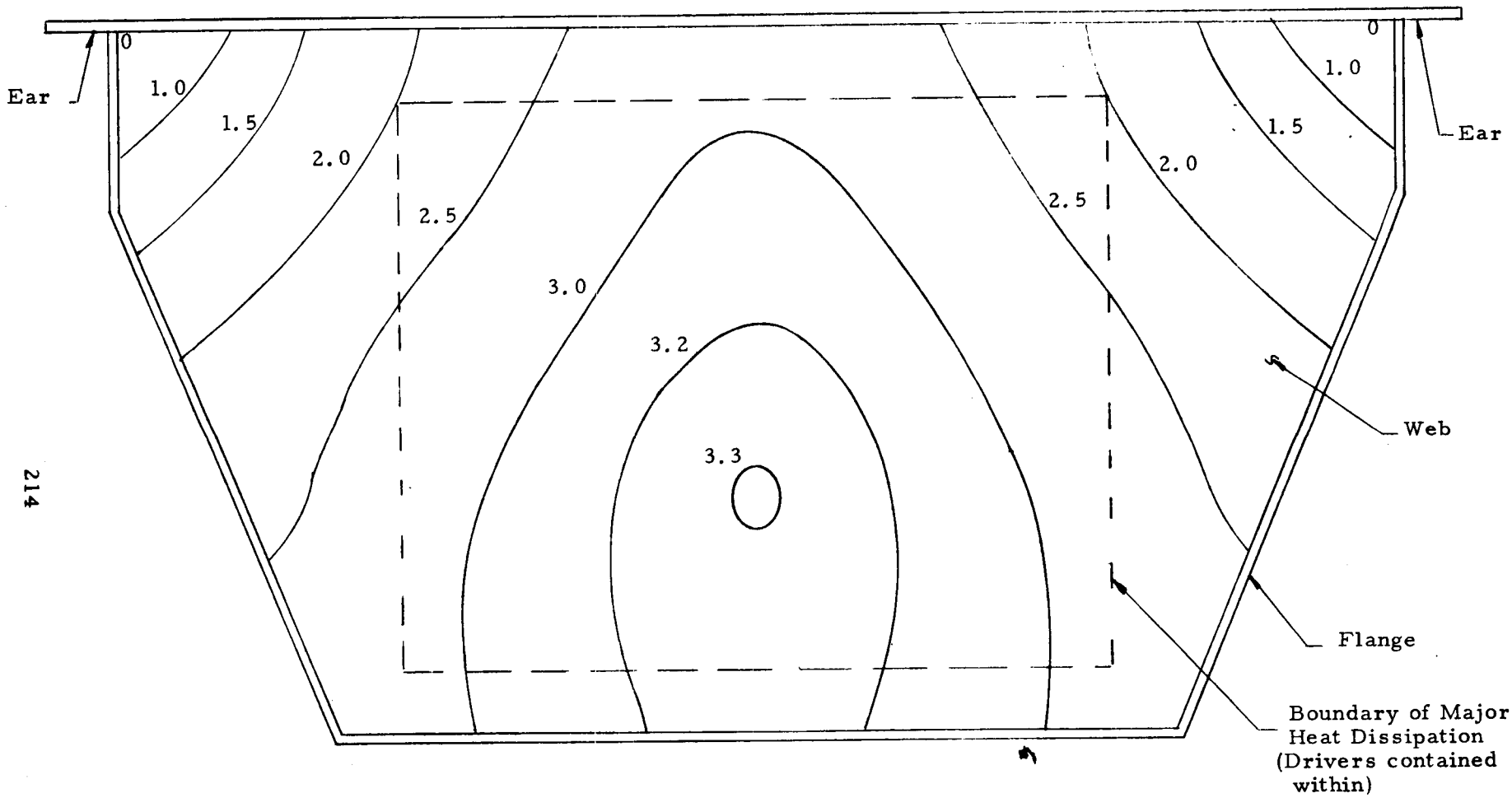
Semiconductor Junction Temperature for 1/32 Inch Aluminum Delta-Pack Frame

	Dissipation Watts	ΔT, Satellite Frame to T_j; °C	Hottest T_j, °C
Memory	1.73	44.5	+74.5
Central Control	1.01**	15.4	+45.4
I-O #1	1.08*	39.8	+79.8
I-O #2	1.06*	12.0	+42.0

*With full component complement ** Some of CCU is in I/O #1.

The major conclusion is that the safety margin between the hottest semiconductor junction and the 100° C limit has been decreased from 32.6°C to 20.2°C (nearly halved).

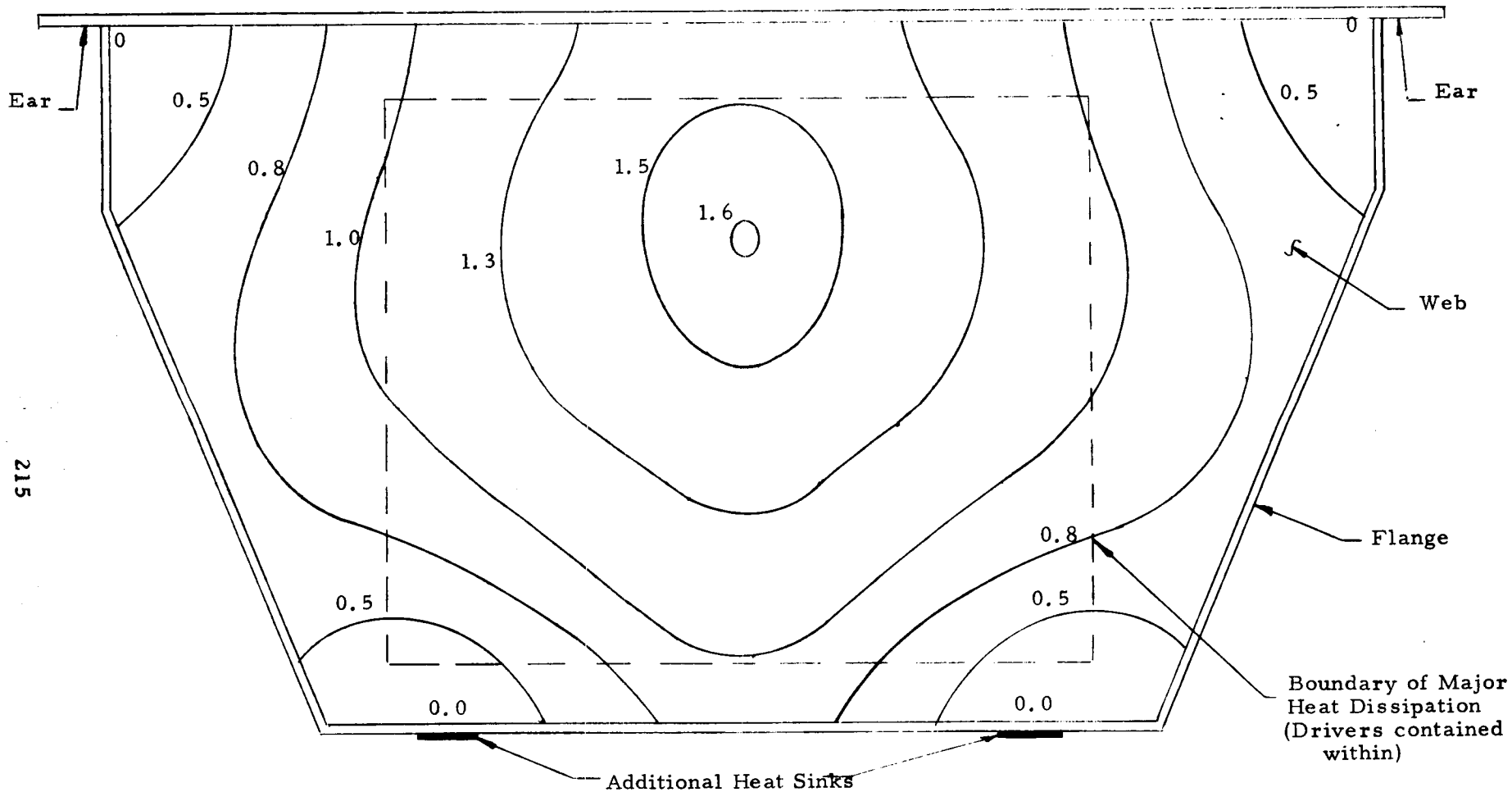
The effect on radiative cooling of the delta-pack subassemblies is to lower the required satellite inner surface skin temperature by approximately 10°C.



NOTE: Dissipation = 1.7 Watts.
Temperatures in Degrees Centigrade Relative to Ear Temperatures.
Web Thickness 3/32 inch, Flange Thickness 1/16 inch, Magnesium Alloy

Temperature Gradient of Memory Delta Pack Frame With Ears Acting as the Only Heat Sink.

Figure 3.7-9



NOTE; Dissipation = 1.7 W.
Temperatures in Degrees Centigrade Relative to Ears and Additional Sinks.
Web Thickness 3/32 Inch, Flange Thickness 1/16 Inch, Magnesium Alloy.

Temperature Gradient of Memory Delta Pack Frame with Heat Sinks at Ears and on Connector End

Figure 3.7-10

3.7.7 Vibration Analysis

In order to provide a delta-pack design with proper vibration characteristics, vibration analyses were performed during the design study. It is important that the delta-pack vibration characteristics be tailored such that when the delta-pack subassemblies are exposed to the anticipated satellite vibration environment, the results be as follows:

1. Components contained on the delta-pack subassemblies are protected from excessive vibration levels.

2. Deflections at delta-pack resonance are minimized to avoid contact between adjacent components and to avoid overstressing printed circuit lines within the MIB's.

3. Stress levels within the delta-pack structure are below fatigue levels.

These goals were realized by mismatching the delta-pack and satellite structure resonant frequencies, and by utilizing materials with high damping capacity for structural members.

Vibration levels created by the Scout Vehicle are defined at the satellite/vehicle interface. These levels, however, will be modified by the satellite structure prior to occurrence at the DPS delta-pack subassemblies. With the assumption that the satellite structure has a fundamental resonant frequency around 100hz, the following random vibration profile was used as a design guide for the DPS delta-pack subassemblies:

0.1 g^2/hz	20-80 hz
0.4 g^2/hz	80-120 hz
0.15 g^2/hz	120-300 hz
0.1 g^2/hz	300-1800 hz

14.1g RMS overall

2 minutes per each of 3 orthogonal axes

Because of the delta-pack geometry the critical axis of vibration is considered to be normal to the MIB's. During vibration in this axis, the major deflecting member will be the MIB's and the supporting delta-pack web, which will deflect as a plate with all edges more nearly simply supported rather than clamped. The flange of the delta-pack frame has high bending stiffness, but inadequate resistance to torsion to completely restrain the MIB's and frame web in a clamped manner. In addition, the simply supported assumption compensates for the small deflection of the frame flange.

The weights of parts contributing to the vibration characteristics of each delta-pack subassembly are listed in Table 3.7-11.

TABLE 3.7-11

EFFECTIVE WEIGHT OF DELTA-PACK SUBASSEMBLIES

	Total weight, lb.	Effective weight, lb.
Memory	1.38	1.16
Central Control	.61	.47
I-O #1	.67	.49
I-O #2	.58	.42

Neither the weight of the structure flange nor flange-mounted connectors were included in the effective weight because they do not affect the vibration characteristics of the MIB and web composite. Parts added to or deleted from the MIB areas will have an approximate effect on the present resonant frequency of the delta-pack subassemblies in a manner inversely proportional to the square root of the increased weight factor.

An approximate calculation for the fundamental resonant frequency of the delta pack is performed by assuming:

1. Equivalent rectangular shape
2. Simply supported edges
3. Uniform weight distribution

The shape of the delta pack is shown in Figure 3.7-11. An equivalent rectangular shape is shown in Figure 3.7-12, with the length L_2 calculated by

$$L_2 = \frac{\int l_2 \, dl_1}{L_1} = 7.83 \text{ inches}$$

where the symbols are defined by Figures 3.7-11 and 3.7-12. The calculation of the fundamental resonant frequency results from the following:

$$f_n = \frac{\omega_n}{2\,Pi} = \frac{Pi}{2}\left(\frac{1}{L_1^2} + \frac{1}{L_1^2}\right)\sqrt{\frac{D}{\mu_1}} \qquad (4)$$

where

$$D = \frac{E_{mib}\left(t_1^3 - t_2^3\right) + E_{web}\, t_2^3}{12\left(1 - \nu^2\right)} \qquad (5)$$

f_n = resonant frequency (HZ)

L_1 = length (7.83 inches)

L_2 = width (5.0 inches)

E_{mib} = modulus of mib (1.8×10^6 psi)

E_{web} = modulus of web (6.5×10^6 psi)

t_2 = plate thickness ($\frac{3}{32}$ inch)

t_1 = total ass'y. thickness ($\frac{7}{32}$ inch)

$_1$ = mass per unit area (varies with type of delta-pack)

ν = poisson's ration ($\approx .3$)

ω = resonant frequency (rad/sec)

Table 3.7-12 lists the resulting resonant frequency for each type of delta-pack.

TABLE 3.7-12

FUNDAMENTAL RESONANT FREQUENCIES OF DELTA-PACK SUBASSEMBLIES

	Frequency, HZ
Memory	465
Central Control	725
I-O #1	710
I-O #2	770

Since the major resonance of the satellite structure is expected to occur around 100Hz, it can be seen that a substantial desirable frequency mismatch occurs, and that the vibration input level to the delta pack will be comparatively low at delta-pack resonance.

The memory plane will have some vibration characteristics of its own. The core-wire mat will be conformally coated to prevent wire abrasion either from the cores or from cross wires.

An unsupported core mat would be expected to behave as a membrane and resonate between 40 and 80 Hz (based on tests of similar planes). To furnish maximum protection to the core-wire mat, particularly from the amplitude and fatigue aspectes, soft foam pads are placed in compression, the core-wire-foam resonance can be raised to around 160 to 200 Hz. This will mismatch the plane resonance with the satellite structure resonance and with the fundamental memory delta-pack resonancy, as well as furnish additional plane protection.

Because of the resonance mismatching approach, it is expected that the overall vibration level to be imposed on such components as flatpacks would not exceed 50g RMS, and might well be much lower. For example, if the transmissibility of the MIB/web assembly (from edge to center) is as much as 20, the overall response at the center is calculated as follows:

$$G_r = \sqrt{\frac{Pi}{2} \; W f_n Q}$$

where

G_r = overall response level (g's RMS)

W = power spectral density near f_n ($.lg^2/Hz$)

f_n = resonant frequency of assembly

Q = transmissibility (20)

Table 3.7-13 lists the resulting overall vibration level for each delta-pack subassembly.

TABLE 3.7-13

VIBRATION LEVELS AT CENTER OF DELTA-PACK SUBASSEMBLIES

	Resonant frequency, HZ	Overall level, gRMS
Memory	465	38
Central Control	725	48
I-O #1	710	47
I-O #2	770	49

Although the vibration response of each delta-pack subassembly is of a random nature, an average or typical deflection value can be calculated. These values for each delta-pack subassembly are listed in Table 3.7-14.

TABLE 3.7-14

AVERAGE DEFLECTION OF DELTA-PACK CENTERS

	Average deflection, single amplitude inches
Memory	0.0023
Central Control	0.00115
I-O #1	0.0012
I-O #2	0.0011

As shown, deflection values are very small and should not cause contact between adjacent parts.

Dynamic stress levels in the delta-pack metal structure and in the MIB's are calculated for the center and for the edge of the delta-packs. In keeping with worst-case philosophy, the equation for the stress level in the delta-pack center is for a simply supported plate, while the equation for the level at the center of the long edge is for a clamped plate. Each equation gives the highest stress level at the location in question.

The basic stress equation for both stress locations is

$$S = \left(\frac{B}{\alpha}\right)\left(\frac{y\,E\,t}{b^2}\right) \qquad (6)$$

where

S = stress in bending

B = constant, depending on case and $\frac{a}{b}$ ratio

α = constant, depending on case and $\frac{a}{b}$ ratio

y = maximum deflection

E = material modules of elasticity

t = plate thickness

b = plate width

a = plate length

For the plate in question, a = 7.83 inches, b = 5.0 inches, and $\frac{a}{b}$ = 1.566. Equation 6 is a modification of the formulas for plate deflections given in Roark, Formulas for Stress and Strain, Fourth Edition, pages 225 and 227, cases 36 and 41. From this reference the values for α and B were obtained corresponding to the $\frac{a}{b}$ ratio. The resulting stresses in the memory delta-pack subassembly (having the highest average displacement of 0.0023 inches) are summarized in Table 3.7-15.

TABLE 3.7-15

AVERAGE DYNAMIC STRESS LEVELS IN MEMORY DELTA-PACK DURING VIBRATION

	Frame web	MIB
$\frac{B}{\alpha}$ center	5.72	5.72
$\frac{B}{\alpha}$ edge	18.75	18.75
E, psi	6.5×10^6	1.85×10^6
t, inches	$\frac{3}{32}$	$\frac{7}{32}$
Stress, center, psi	320	210
Stress, edge, psi	1050	700

These levels are very low for both the metal web and the MIB surface.

If the delta-pack structure web and flange thickness were decreased to $\frac{1}{32}$ inch and the metal changed to an aluminum alloy, the vibration characteristics of each delta-pack would be changed due to changes in weight (decrease) and plate stiffness (decrease). The change in effective weight varies from 5% to 15%; however, the change in plate stiffness amounts to a decrease ot 69%, primarily because the effective load carrying members, the MIB's, are placed closer to the neutral axis of the web/MIB composite. Taking into account only the changes of weight and stiffness, the fundamental resonant frequencies are changed as shown in Table 3.7-16.

TABLE 3.7-16

DELTA-PACK RESONANT FREQUENCIES WITH A FREAM WEB OF 1/32 INCH ALUMINUM

	Resonant frequency, HZ
Memory	265
Central Control	435
I-O #1	425
I-O #2	465

By comparing the data of Table 3.7-16 to that of Table 3.7-12, it is noted that the fundamental resonant frequencies have been decreased by several hundred HZ. Of particular significance is the changed resonant frequency of the memory delta-pack, which now lies in a higher range (.15 g^2/HZ vs 0.10 g^2/HZ) of the design guide vibration profile. In addition, the new frequency approaches the resonant frequency range of the memory plane causing the plane to endure a higher vibration environment.

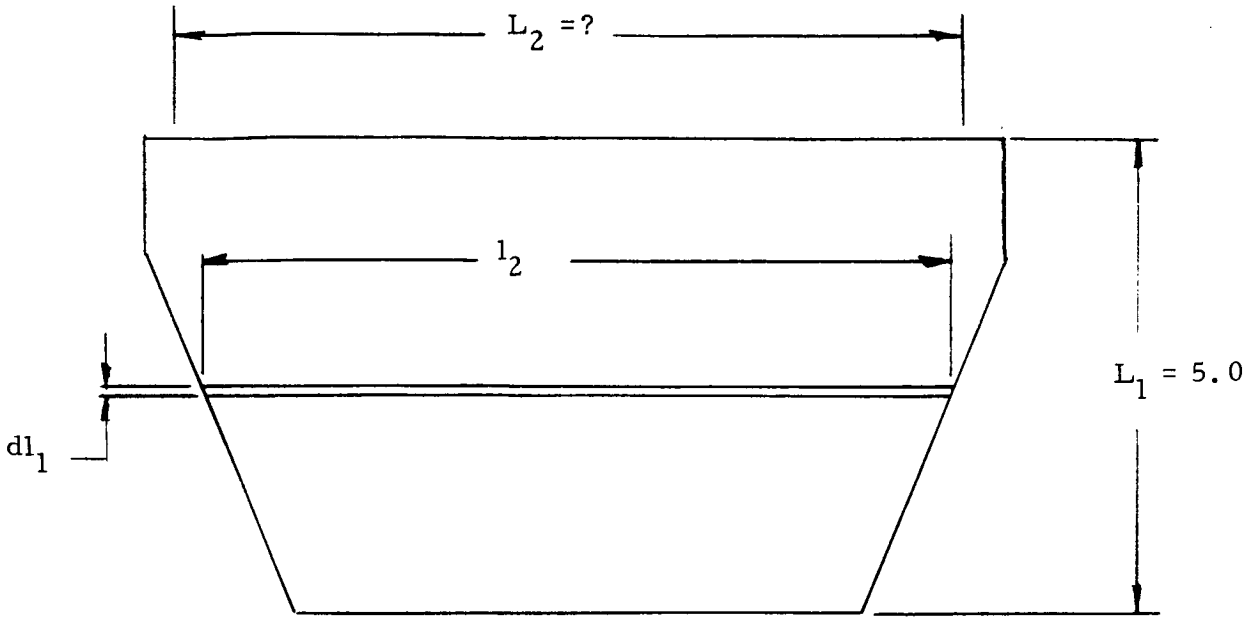


Figure 3.7-11

Illustration of Nomenclature for Calculating the Equivalent Rectangular Shape of Delta-Pack Frame Web.

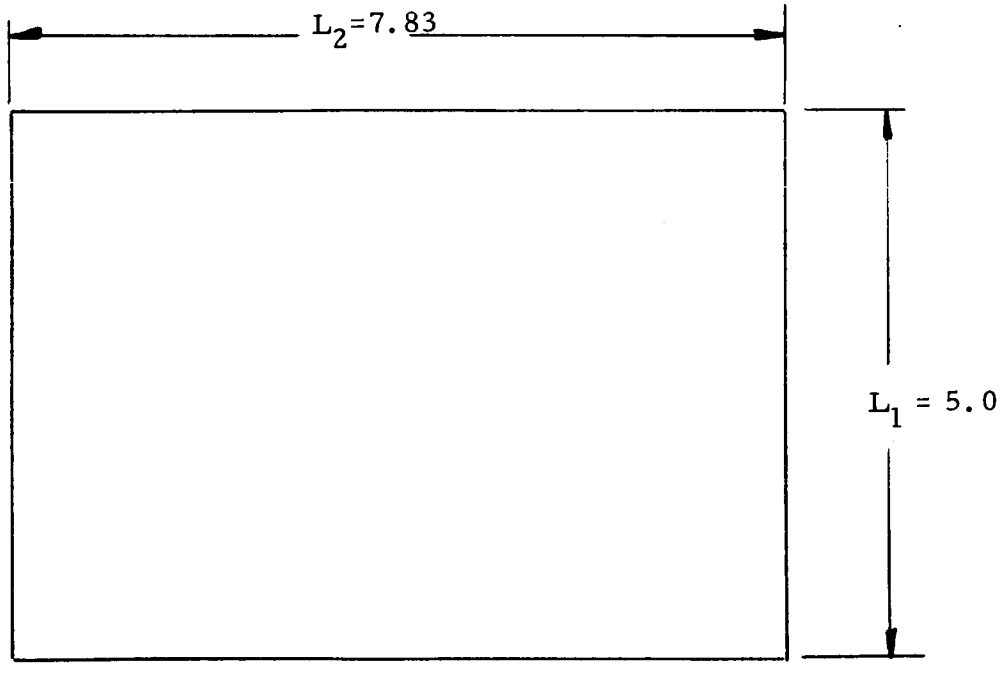


Figure 3.7-12

Equivalent Rectangular Shape of Delta-Pack Frame Web.

3.7.8 Magnetic Cleanliness

Study of the magnetic shielding requirements for the S^3 program indicate that the shielding design will be dependent on such a large number of variables that analytical study, without extensive breadboarding and tests, would yield inconclusive results.

The problem of keeping the magnetic field at the experiment sensors at some minimum level appears to be a problem of overall satellite system design rather than of the data processing system design alone.

The location of the data processing system with respect to the experimenter's magnetometers is obviously a large contributing variable. It would be preferable to locate the data system on the side of the satellite opposite the magnetometers, thus taking advantage of nearly 30 inches of distance and the metal shielding of the satellite frame.

The requirement of a magnetic field of not more than 2.5 gamma at a distance of three feet from the data processing system does not specify direction, which is an important contributing factor. From the drawings of the satellite shown in the NASA S^3 Feasibility Study, it may be assumed that the data system delta packs will present an edgewise aspect to the magnetometers. This would imply that shielding should be concentrated about the metal delta pack frame. However, with varying experiments, the aspect may change.

As the frequency of the magnetic field increases, it becomes simpler to shield against it. Laminations of high permeability materials such as Hipernik or Hymu 80 alternated with copper would effectively shield the steep wavefronts generated by fast current pulses. As the frequency of the fields approaches zero, the only effective shield appears to be high permeability material alone, surrounding the fields in order to provide a closed flux path.

The data processing system logic delta packs will probably not be a serious source of magnetic interference. Logic and I/O circuit currents will be so low because of the low power DTL and MOSFET circuits that the Kovar cases of the flatpack units should provide effective shielding. Kovar has fairly high permeability and low remanence which are desirable magnetic shielding qualities.

The memory delta pack will probably present the greatest shielding problem because of the high drive currents and steep wavefronts of the drive pulses. However, since the duration of the drive pulses is short, the problem may resolve itself mainly into one of high frequency shielding by laminar techniques. As an initial approach, the magnesium alloy end caps on the memory core array may be replaced by laminated shielding material. This should shield against magnetic pulses generated by the coordinate selection and inhibit currents. The wiring on the printed circuit boards will be arranged to keep current carrying loops as small as possible. The discrete wires carrying the power supply and return wires between the printed circuit board and the delta pack power connector will be twisted to minimize current loops.

Outside the delta packs, the power conductors in the center of the satellite may be a greater source of magnetic interference than the delta pack circuits. Careful cable design with all current carrying leads and returns twisted and shielded will be important in this area.

The satellite structure surrounding the data system delta packs will provide some degree of shielding for the higher frequency magnetic fields.

It must be reemphasized that the magnetic cleanliness aspect should be approached from an overall system viewpoint, otherwise there is likely to be much duplication of effort in shielding designs, with a consequent addition of weight to the satellite.

3.8 Reliability

The reliability analysis performed on the S^3 study is discussed in the following subsections. Included is a discussion of the system reliability for complete mission success and for success with a minimal useful capability, and failure rates for components with application ground rules and data sources.

3.8.1 System Reliability Analysis

For the S^3A the probability of no failures for one year of continuous operation is 0.7939. The probability of successfully obtaining all data, though in a modified manner, is 0.8365. The probability of obtaining all analog data is 0.8754 and of all accumulator data is 0.8563. However, the probability of losing a single accumulator, for example, is only 0.0054. These figures are predicated on a remaining useful amount of main store and processor capability in the event of a failure. Also included is the capability to be able to process all input data properly and to be able to present data to the tape or transmitter properly with proper message headers. The probability of success for the hardcore for one year is 0.8927; this is the probability of no catastrophic failures.

For the minimum system, assuming 2K of main store, A/D converter, compressor, and 16 channels, the probability of no failures for one year of continuous operation is 0.8347. The probability of successfully obtaining all data though in a modified manner is 0.8715. The probability of obtaining all analog data or all accumulator data is 0.8807

Table 3.8-1

S^3 Reliability Estimates (For Successful Total Capability, Data Return) and Hardcore

S^3 With 2K Byte Mainstore

Configuration / Unit	Probability of Success		
	Total Capability	All Data Return	Hardcore
Oscillator	.9950	.9950	.9950
Central Control	.9199	.9499	.9499
2K byte Mainstore	.9610	.9716	.9716
Input/Output	.9490	.9490	.9770
System Reliability	.8347	.8715	.8972

S^3 A With 4K Byte Mainstore

Configuration / Unit	Probability of Success		
	Total Capability	All Data Return	Hardcore
Oscillator	.9950	.9950	.9950
Central Control	.9199	.9499	.9499
4K byte Mainstore	.9479	.9673	.9673
Input/Output	.9150	.9150	.9764
System Reliability	.7939	.8365	.8927

and 0.8915, respectively. As with the S^3A these figures are predicated on a remaining useful amount of main store and processor capability and on being able to process format and output data properly. For the hardcore, the probability of success is 0.8972; this is the probability of no catastrophic failures in a year of operation.

Table 3.8-1 contains the S^3 reliability estimates for successful total capability, data return, and hardcore for the two system configurations discussed.

The reliability model used for the system for total mission capability is based on the constant component part failure rate concept. The assumption is made that component parts are failure independent and that any part failure constitutes an equipment failure.

Figure 3.8-1 depicts the reliability model for the system. Within each of the units (except the oscillator) a portion of the hardware is designated hardcore in that a failure would prevent proper system operation or result in erroneous data. The remainder is non-catastrophic in that it can be bypassed or affects only one channel or type of data. Individual areas of the S^3 system are discussed in detail in the following sections.

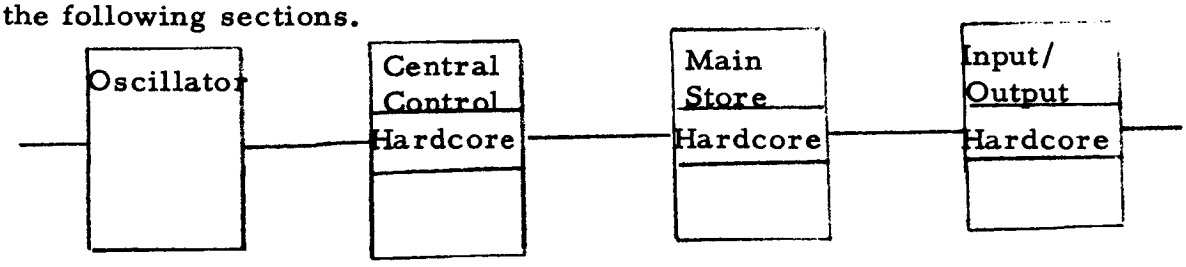


Figure 3.8-1

S^3 Reliability Model

3.8.1.1 Oscillator

The oscillator failure rates are described in Table 3.8-2. A component part failure in the oscillator will result in an oscillator failure which will prevent system operation, hence the oscillator is considered to be completely hardcore. Oscillator reliability for the specified mission is .9950.

3.8.1.2 Central Control

The failure rates for the Central Control are described in Table 3.8-3 for the hardcore and non-catastrophic components. Reliability of the Central Control hardcore was computed to be .9499 while the Central Control reliability for total S^3 capability is .9199. Catastrophic (hardcore) and non-catastrophic conditions for the Central Control are discussed below.

The Central Control has a number of alternate modes of operation and redundant control capabilities. Many missions will not require the full capabilities of the system and all mission requirements can be satisfied to some extent in degraded system modes. A degraded mode is one where a part of the system is non-operative and it is replaced with some combination of hardware and software bypass. For example, if a failure or several failures should eliminate the usefulness of the DSC as a system clocking function, a ground control could easily switch the clock generator to the SCC mode. Now there would be an operable clock but it would not be collecting data in exact synchronization with the spin rate and some reprogramming might be desired.

Table 3.8 - 2

Oscillator Failure Rates

Components	Quantity n	Failure Rate λ (Failures/Million hrs.)	$n\lambda$
Resistors	4	.01	.040
Diodes	1	.0006	.001
Capacitors	3	.001	.003
Transistors	1	.00086	.001
Crystal	1	.5	.500
Integrated Circuit 9041	1	.0125	.013
Connections (Solder)	35	.0001	--
MIB	--	--	.015
Failure Rate (Failures/Million Hours)		$\sum n\lambda_{osc}$	.573

$$R_{OSC} = e^{-\lambda T} = e^{-(.573 \times 10^{-6})(8760)} = e^{-(.0050)} = .9950$$

Table 3.8-3

Central Control Failure Rates

Central Control Components	λ Failures/ Million Hrs.	Hardcore		Non-catastrophic	
		n	n λ	n	n λ
Integrated Circuits					
9040	.0244	128	3.123	67	1.635
9043	.0147	4	.059	2	.029
9044	.0147	8	.118	9	.132
9046	.0168	25	.420	12	.202
9047	.0189	2	.038	18	.340
9048	.0168	25	.420	14	.235
9049	.0189	2	.038	4	.076
9930	.0168	5	.084	2	.034
9933	.0053	3	.016	2	.011
Connections Solder	.0001	2828	.283	1820	.182
MIB's and Connectors	--	--	1.234	--	.794
Central Control Unit $\sum_n \lambda$	Failures / Million Hrs		5.833		3.670

$$R_{CCU\text{-}Hardcore} = e^{-\lambda_{H.C.} T} = e^{-(5.833 \times 10^{-6})(8760)} = e^{-(.0515)} = .9499$$

$$R_{CCU\text{-}Total\ Capability} = e^{-\lambda_{C\varphi} T} = e^{-(9.503 \times 10^{6})(8760)} = e^{-(.0837)} = .9199$$

In generating reliability estimates, two separate failure probabilities are calculated. The first, is the catastrophic failure rate which considers the hardcore--that part of the Central Control absolutely necessary for the system to stay operable. The second is based on the probability of failures which will put the machine into some slightly degraded mode of operation. The failures which are allowable cannot be easily described as there are numerous combinations of multiple failures which will not interact to further degrade the operational capability.

Definition of the hardcore of the Central Control is necessary for a valid derivation of the catastrophic failure rate. It is, perhaps, more meaningful to explain that part of the Central Control which is not a part of the hardcore and to define alternate modes of operation.

In the clock generation area, the entire DSC generation area can be eliminated. When the DSC is not working, the SCC must be available in its place. The control timing is quite necessary regardless of which system clocks are used.

In the input area, neither the Discrete Input Register nor the Mode Code register is hardcore, although it is highly improbable that all stages of both registers would fail at the same time. Still, lacking these registers, the program could sense external conditions through the inputs to the Conditional Branch and Wait instructions. Reprogramming would be required to adapt to these failures as is the case

with most degraded modes.

The Maximum Buffer Register (MXB) and Comparator could be eliminated with a slight increase in software requirements to program the resetting of the Input Address Register when the Buffer is filled so that data won't be written over the program. Further degradation of the system, in event of loss of the MXB, would be evident in the output data which will include the program every time the output sequences through the memory, because it will go through the entire memory rather than recycle to zero every time the Buffer area has been read.

The entire indexing capability could be eliminated with very little degradation in the system capability but with a major increase in the program storage requirements. If the storage capability were already seriously taxed with the mission experiment data sampling requirements, then these requirements would have to be relaxed to allow for the decreased data Buffer size. Eliminating the indexing capability includes the Index Register and associated gating, the Index Register Address Generator, the Set Index instruction, and the Branch on Index conditions of the Branch instruction.

The Wait instruction could also be eliminated. This instruction can be easily replaced with a one step looping Branch instruction. In this case, there is no degradation in system capability and an insignificant increase in program memory requirements. However, the Memory power requirements will be increased by 1.4 watts during

the time the machine is in a high speed loop rather than in an idle state.

The Set Discretes instruction is probably necessary for experiment control on most missions. The elimination of a partial capability of this instruction would be allowable but alternate modes and perhaps higher sampling rates would be required to compensate for this lack of control.

The Route, Branch, and Fetch instructions are all hardcore. A small partial capability of each instruction (such as some of the Branch conditions) may be lost without serious system degradation. Partial capability of this sort is not considered in the reliability estimates; the whole is considered hardcore. The remaining control areas and registers which have not yet been discussed are also necessary to system operation and are considered to be fully a part of the hardcore.

3.8.1.3 Main Store

Two mainstore configurations were evaluated for the S^3 system, a 2K byte and a 4K byte. Failure rate estimates for the main store devices with complete capability are shown in Table 3.8-4. The reliability estimates for the 2K byte and 4K byte mainstore devices, with no failures for a one year mission, are .9610 and .9479 respectively.

The reliability estimate (based on a one year mission) for the 2K byte main store hardcore is 0.9716. For the 4K byte main store hardcore, the reliability estimate is 0.9673.

Main store hardcore for the purposes of this analysis has been defined as at least three-fourths of a main store device operating

Table 3.8-4

Mainstore Failure Rate Estimates

Mainstore Components	Failure Rate λ Failures / Million Hours	2K Byte		4K Byte	
		n	n λ	n	n λ
Dual Transistors	.00258	45	.116	51	.132
Integrated Circuits	--	--	--	--	--
DTμL 932	.062	4	.248	4	.248
DTμL 946	.0125	16	.200	18	.225
DTμL 951	.014	13	.182	13	.182
μA 711	.0191	4	.076	4	.076
LPDTμL 9041	.0125	28	.350	32	.400
Resistors	.00263	230	.605	266	.700
Potentiaometer	.08	7	.560	7	.560
Temperature Sense Element	.01	1	.010	1	.010
Diodes, Switching	.0006	6	.004	6	.004
Diodes, Zener	.00085	28	.024	33	.028
Diodes, (16 Diodes/Flatpack)	.0099	28	.277	32	.317
Capacitors, Ceramic	.01	16	.160	16	.160
Capacitors, Tantalum	.034	1	.034	1	.034
Page MIB, holes	.000428	2,415	1.034	2,753	1.178
Page MIB, Segments	.000017	1,208	.021	1,376	.023
Connector, Pin Pair	.000126	80	.010	80	.010
Connector, Body	.005	2	.010	2	.010
Tape Cable Lines	.0017	80	.136	80	.136
Jumpers	.00002	192	.004	256	.005
Solder, Tape Lines	.001	160	.160	160	.160
Solder, Jumpers	.00054	384	.207	512	.276
Solder, Connectors	.00054	160	.086	160	.086
Solder, Components	.0001	2,415	.242	2,753	.275
Array	--	--	--	--	--
Cores	.00004	7,936	.317	15,872	.635
Wires	.0001	196	.020	260	.026
Welds	.0001	392	.039	520	.052
Mainstore $\sum n\lambda$	Failures / Million Hours		4.934		5.948

$$R_{\text{Mainstore 2K}} = e^{-(4.934 \times 10^{-6})(8760)} = e^{-(.0390)} = 0.9610$$

$$R_{\text{Mainstore 4K}} = e^{-(5.948 \times 10^{-6})(8760)} = e^{-(.0521)} = 0.9479$$

successfully during a mission. A bit position failure in all words, of course, is a total failure; but inability to properly address several words or bytes is not a total failure and can be programmed around. Analysis has shown that several failures may occur within a main store and not exceed the defined hardcore capability. Items of significance included in this area are the EI drivers, diode matrix, magnetic cores, connections and wires. Table 3.8-5 contains a tabulation of the failure rates for these items.

The reliability of one EI driver was calculated to be .9997. In the analysis of the main store devices a worst case condition was indicated for the 2K byte. The 2K byte has 28 EI drivers, should 1 EI driver of a particular 4 fail or 2 EI drivers of the remaining 24 fail, the main store would degrade to the defined minimum hardcore capability. The expression for the probability of success of the EI drivers for minimum hardcore capability is then:

$$R_{\text{EI Drivers}} = R^4(R^{24} + 24R^{23}(1-R) + 276R^{22}(1-R)^2) + R^{24}(4R^3(1-R))$$

by substitution $R = R_{EI} = .9997$

then $R_{\text{EI Drivers}} = .999991$

Since the probability of success for the EI drivers is quite high (.999991) compared to the probability of success for the total main store, hardcore reliability estimates are simplified by considering the failure rate of the EI driver as part of the non-catastrophic failure rate. This approach is also followed in the analysis for the cores, diode matrix, connections and wires. For these cases, several hundred cores or

Table 3.8 - 5

Failure Rates - EI Driver, Cores, Diode Matrix, Connections and Wires

A. Computation of Failure Rate for 1 EI Driver

EI Driver	Failure Rate - λ (Failures/Million Hours)	Quantity n	$n\lambda$ Failures/Million Hrs.
Dual Transistors	.00258	1 1/2	.004
DTμL 946	.0125	1/2	.006
LPDTμL 9041	.0125	1	.013
Resistors	.00263	3	.008
Diode, Flat Pack	.0099	1/8	.001
Diodes, Zener	.00085	1	.001
Connections, Solder	.0001	42	.004
	$\Sigma n\lambda$ (Failures/Million Hours)		.037

B. Failure Rate Contribution of EI Drivers, Cores, Diode Matrix, Connections and Wires to Total Mainstore Failure Rate

Component	Unit Failure Rate (Failures/Million Hours) 2K Byte		4K Byte	
EI Drivers	(28)	1.036	(32)	1.184
Cores		.317		.635
Diode Matrix		.277		.317
Connections and Wire		.060		.080
$\Sigma n\lambda$ (Failures/Million Hours)		1.690		2.216

Calculation of the reliability of an EI driver is as follows:

$$R_{EI} = e^{-\lambda_{EI}T} = e^{-(.037 \times 10^{-6})(8760)} = e^{-(.0003)} = .9997$$

connections must fail to fail the system or several diodes in a diode matrix or several wires within the core array. The probability of several of these occurring is even more remote than a catastrophic failure due to EI drivers. With this reasoning, the failure rates compiled in Table 3.8-5B are considered to be non-catastrophic. Reliability calculations for main store hardcore for both 2K and 4K byte configurations were performed as follows:

Main Store Hardcore Reliability Calculations

	2K Byte	4K Byte
FailureRate Total	4.934	5.948
Failure Rate Non-catastrophic	-1.690	-2.216
Failure Rate Hardcore	3.244	3.732

2K byte Hardcore Reliability

$$R_{2K\text{-}HC} = e^{-\lambda_{2K\ HC}T} = e^{-(3.244 \times 10^{-6})(8760)} = e^{-(.0284)} = .9716$$

4K byte Hardcore Reliability

$$R_{4K\text{-}HC} = e^{-\lambda_{4K\text{-}HC}T} = e^{-(3.732 \times 10^{-6})(8760)} = e^{-(.0327)} = .9673$$

3.8.1.4 Input/Output

The S^3A reliability estimates are contained in Table 3.8-6A. For total I/O capability, the probability of successful operation without failure for a one year mission is 0.9150. For the common hardcore

which includes word length, control counter, addressable multiplexor and message header hardware, the probability of success is .9764. The probabilities of success for the A/D converter and compressor are .9985 and .9947 respectively. The probabilities for successfully obtaining all Analog or all Accumulator Data are .9580 and .9366 respectively. (The probability of a single accumulator failure is 0.0054, see Table 3.8-7.) Probability of a single accumulator failure is $(1-R_{success})$.

For the minimal system, with one Discrete Output Register, 24 channels, A/D converter, compressor, and 2K of main store, the reliability estimates are given in Table 3.8-6B. In this case the probability of successful operation of the total I/O without failure is 0.9490. For the common hardcore, the probability of success is 0.9770. The probabilities of success for the A/D converter and Compressor are 0.9905 and 0.9947, respectively. The probabilities of successfully obtaining all Analog or all Accumulator Data (no accumulators in the estimate) are .9591 and .9708, respectively.

3.8.2 Failure Rates, Ground Rules and Sources

The component part and connection failure rates, application ground rules and a general discussion of sources are contained in the following sections.

Table 3.8-6A
S^3A Input/Output Failure Rates and Reliability Estimates

Component Type	Failure Rate Failures / Million Hrs.	Total I/O		Common Hardcore		A/D		Compressor		Analog Data		Accumulator Data	
		n	$n\lambda$	n	$n\lambda$	n	$n\lambda$	n	$n\lambda$	n	$n\lambda$	n	$n\lambda$
9040	.0244	184	4.489	44	1.074	10	0.244	14	0.342	54	1.318	158	3.855
9042	.0126	11	0.139	7	.088	--	--	--	--	7	.088	7	.088
9044	.0169	12	.203	10	.169	--	--	1	.017	11	.186	11	.186
9046	.0168	14	.235	6	.101	--	--	4	.067	6	.101	11	.185
9047	.0189	13	0.246	4	.076	--	--	1	.017	8	.151	9	.170
9048	.0168	63	1.058	25	.420	6	.101	--	--	37	.622	59	.991
9933	.0053	12	.004	7	.037	--	--	1	.005	7	.037	8	.042
Transistors	.00086	14	.012	--	--	4	.003	--	--	14	.012	--	--
Resistors	.00263	14	.037	--	--	4	.011	--	--	14	.037	--	--
MEM 2009	.029	2	.058	--	--	--	--	--	--	2	.058	--	--
MOS 810	.551	1	.551	--	--	1	.551	--	--	1	.551	--	--
Others*	.674	1	.674	--	--	--	--	--	--	1	.674	--	--
Connections, Solder	.0001	4464	.446	1442	.144	284	.028	294	.029	2103	.210	3682	.368
MIB's and Connectors	--	--	1.949	--	.629	--	.124	--	.128	--	.918	--	1.607
$\Sigma n\lambda$, Failures / Million Hours			10.161		2.738		1.052		0.605		4.914		7.492
Reliability (1 year mission)			.9150		.9764		.9985		.9947		.9580		.9366

*See Table 3.8-6C

Table 3.8-6B Minimum System I/O Failure Rate and Reliability Estimates

Component Type	Failure Rate Failures / Million Hrs.	Total I/O		Common Hardcore		A/D		Compressor		Analog Data		Accumulator Data	
		n	nλ	n	nλ	n	nλ	n	nλ	n	nλ	n	nλ
9040	.0244	76	1.854	44	1.074	10	0.244	14	0.342	54	1.318	58	1.415
9042	.0126	8	0.101	7	.088	--	--	--	--	7	.088	8	.101
9044	.0169	12	.203	10	.169	--	--	1	.017	11	.186	11	.186
9046	.0168	9	.151	3	.050	--	--	4	.067	3	.050	8	.134
9047	.0189	13	0.246	4	.076	--	--	1	.019	8	.151	9	.170
9048	.0168	36	.605	25	0.420	6	0.101	--	--	37	0.622	23	.386
9933	.0053	9	.048	7	0.037	--	--	1	.005	7	.037	9	.048
Transistors	.00086	7	.006	--	--	4	.003	--	--	7	.006	--	--
Resistors	.00263	7	.018	--	--	4	.011	--	--	7	.018	--	--
MEM 2009	.029	1	.029	--	--	--	.029	--	--	1	.029	--	--
MOS 810	.551	1	.551	--	--	1	.551	--	--	1	.551	--	--
Others*	.674	1	.674	--	--	--	--	--	--	1	.674	--	--
Connections, Solder	.0001	2486	.249	1400	.140	284	.028	294	.029	1982	.198	1764	.176
MIB's and Connectors	--	--	1.242	--	.611	--	.122	--	.128	--	.858	--	.771
$\sum n\lambda$ Failures / Million Hours			5.977		2.665		1.089		0.607		4.786		3.387
Reliability (1 Year Mission)			.9490		.9770		.9905		.9947		.9591		.9708

*See Table 3.8-6C

Table 3.8-6C I/O Other - Failure Rate Estimates

Component Parts	Failure Rate (Failures/Million Hrs)	n (Quantity	$n\lambda$ (Failures/Million Hours)
Resistors, MF	.00263	22	0.058
Resistors, Variable	.08	3	0.240
709	.016	2	0.320
Diode, Zener	.0085	2	0.017
Transistors, Dual	.00258	2	0.005
Capacitor, Tantalum	.034	1	0.034
$\Sigma n\lambda$ (Failures/Million Hours)			0.674

Table 3.8-7 Accumulator Failure Rate Estimate

Component Part	Failure Rate $\left(\frac{\text{Failures}}{\text{Million Hours}}\right)$	n (Quantity)	$n\lambda$ (Failures/Million Hours)
9040	.0244	20	.488
9048	.0168	6	.101
Connections, Solder	.0001	364	.036
$\Sigma n\lambda$ $\frac{\text{Failures}}{\text{Million Hours}}$			.625
Reliability (1 year mission)			.9946

3.8.2.1 Failure Rates

The failure rates used in estimating unit reliabilities are presented in Table 3.8-8 General ground rules associated with these are listed below; specific ground rules are mentioned under the "Comments" section of the Table.

- Component part hot spot temperatures are not greater than 65°C
- All component parts are operated at ≤ 50% of the manufacturer's rating (in addition to the above constraint on hot spot temperature)
- Component parts are procurred for 1968 production hardware
- Failure rates are applicable to on-orbit environmental conditions in well packaged hardware
- The failure rates are achievable under a high reliability program as defined by IBM. This program includes such requirements as:
  - Regulated procurement specifications for all parts
  - Training, certification and audit of operators for critical processes
  - Lot control of all critical parts
  - 100% inspection of all solder and weld connections at ≥10 power magnification
  - 100% inspection of critical part parameters
  - 100% mechanical screening as required.

3.8.2.2 Failure Rate Sources

The component part failure rates on well-established discrete parts such as resistors, capacitors, and semiconductors are based primarily upon observed field failure data obtained from the ground operation similar parts in ESC-manufactured Missile Guidance and space vehicle computers and their associated ground equipment. The field failure rate data for the Missile Guidance Computers are included in the FARADA Handbook (Naval Bureau of Weapons) under Source No. 204 and represent over 1.0 billion component part operational hours of data for resistors, capacitors, diodes, and transistors where equipment operate times and failure verifications are carefully monitored. Other sources of data include IBM's Saturn V Computer and Data Adapter, the Gemini computer, IBM commercial computers (representing over 270 billion component part operational hours). Minuteman field data (over 180 billion part hours), IDEP, FARADA (over 2,600 billion part hours), ECRC, supplier and IBM part life tests and all other programs over which IBM-ESC has surveillance.

In cases where neither observed data nor reasonable predicted rates are available, failure rates are synthesized, based upon a summation of the failure rates of its elements. Observed failure data are extrapolated as necessary, based on engineering judgment, for new or different conditions, materials, designs, environments, or stresses. MIL Handbook 217 and RADC Reliability Notebooks are, on occasion, used as guidelines.

Table 3.8-8

High Reliability Program Failure Rates for Ground/Orbit Environment, 1968 Production

Component Part/Connection		Failure Rate (Failures/ Million Hours	Comments
Integrated Circuit and Discrete Semiconductors			Procurement of Integrated Circuit and Discrete Semiconductors includes:
• Integrated Circuits	9040	.0244	1. 100% electrical inspection
	9041	.0125	2. Impact shock, 100%
	9042	.0126	3. Constant acceleration, 100%
	9043	.0147	4. X-Ray 100%
	9044	.0169	5. Burn in 100%
	9046	.0168	6. Electrical and microscopic destructive analysis of samples from each lot
	9047	.0189	
	9048	.0168	7. Lot analysis testing
	9049	.0189	8. Resident or roving auditor to analyze vendor test results and maintain process control charts.
	9930	.0168	
	9933	.0053	
	932	.062	
	946	.0125	
	951	.014	
	711	.0191	
	816	.551	
	2009	.029	
	709	.016	
• Transistors, Switching		.00086	
Transistors, Power		.00136	
Transistors, Dual		.00258	
• Diodes, Switching		.0006	
Zener		.0085	
16/Flatpack)		.0099	
Resistors, Metal Film		.00263	100% electrical and visual inspection, x-ray and environmental screening tests dependent on part type and vendor
Resistors, Variable		.08	
Capacitors, Ceramic		.01	100% electrical inspection of capacitance, dissipation factor, insulation resistance, dielectric strength
Capacitors, Tantalum		.034	

Table 3.8-8 (Cont'd)

Component Part/Connection	Failure Rate (Failures/ Million Hours	Comments
Cores (Memory)	.00004	
Crystal	.5	100% visual and electrical inspection at vendor and IBM, 500 hour burn-in
Connector, Per Pin Pair	.00126	100% visual inspection
Connector, Per Body	.005	100% visual inspection
Multilayer Interconnection Board		100% electrical, visual and x-ray inspection of holes; special tests, i.e., microsectioning performed on test holes from each board
Per Landed Hole	.000428	
Per Conductive Segment	.000017	
Tape Cable, Per Line	.0017	100% electrical testing
Harness Wire, Per Line	.0017	100% electrical testing
Solder Joint, Components and Flatpacks	.0001	100% inspection at $\geq$ 10 power magnification; conformal coating to immobilize parts
Solder Joint, Tape Cable Terminator	.001	100% inspection at $\geq$ 10 power magnification
Solder Joint, Connectors and Jumpers	.00054	100% inspection at $\geq$ 10 power magnification, encapsulated to restrict motion
Welds	.0001	100% inspection at $\geq$ 10 power magnification

3.9 Programming Applications

This section is a general guide to programming the S^3 Data Processing System. It illustrates some of the more direct applications of the instruction set to the solution of typical problems.

The S^3 instruction set and memory organization provide the capabilities to perform many functions normally provided only by a much larger instruction repertoire. The instruction set, although very powerful and versatile, is small in number and therefore readily learned and easily programmed. These features provide the program flexibility and expandability necessary to satisfy the different experimental missions.

The instruction set provides for: processing a single byte or multiple bytes of data, from one channel or multiple channels and at a burst rate or clock sequential rate, routing the data either to the tape for storage or to the transmitter for real time transmission, communicating with the outside world by the input and output discretes, sensing various conditions, looping, indexing, linkage and branching (conditional and unconditional).

The instruction set consists of FETCH, ROUTE, BRANCH, SET DISCRETES, SET INDEX and WAIT as are defined in section 3.2.

This section illustrates the capabilities of the S^3 instruction set by applying it to general software examples and combined program example.

3.9.1 General Programming Examples

3.9.1.1 Acquiring Accumulator Data

Problem: Data must be simultaneously accumulated at least 80% of the time in five, 20-bit Accumulator registers. This data must be collected 32 times each revolution of the satellite, which has a 4 RPM spin rate, and then compressed into 9 bits (3 bytes). Also, the modes of the sensors providing the pulse inputs must be changed each 1/2 rev.

Solution: The collection of the data can be done by one fetch instruction with the following format:

	Starting Channel	Number bytes per channel	Chain	Repeat 5 times and increment channel address each time
• FETCH	1	3	CC	5I

This instruction will cause 3 bytes (12 bits) of data to be read from each of 5 channels starting with channel 1. The CC will cause the repeat fetches to be executed in a chain mode, that is, execution takes place as soon as the channel has transferred out the last byte of data for the previous execution.

During the actual data fetching, the accumulation of data must be terminated. This is accomplished by closing the accumulator gates before the fetch and then reopening the gates upon completion of the fetch. This gives:

• DISCRETE	CLOSE GATES
• FETCH	1, 3, CC, 5I
• DISCRETE	OPEN GATES

Since the data is being fetched many times during a revolution and at regular intervals, a timing control of the fetching is necessary. This is done by the WAIT instruction which causes an idle machine time until the condition specified is met. In this illustration, a pulse or signal which occurs 32 times each revolution is desirable. This signal is labeled as the Sector pulse and the instruction sensing this pulse is placed preceding the previous routine and is:

- WAIT S

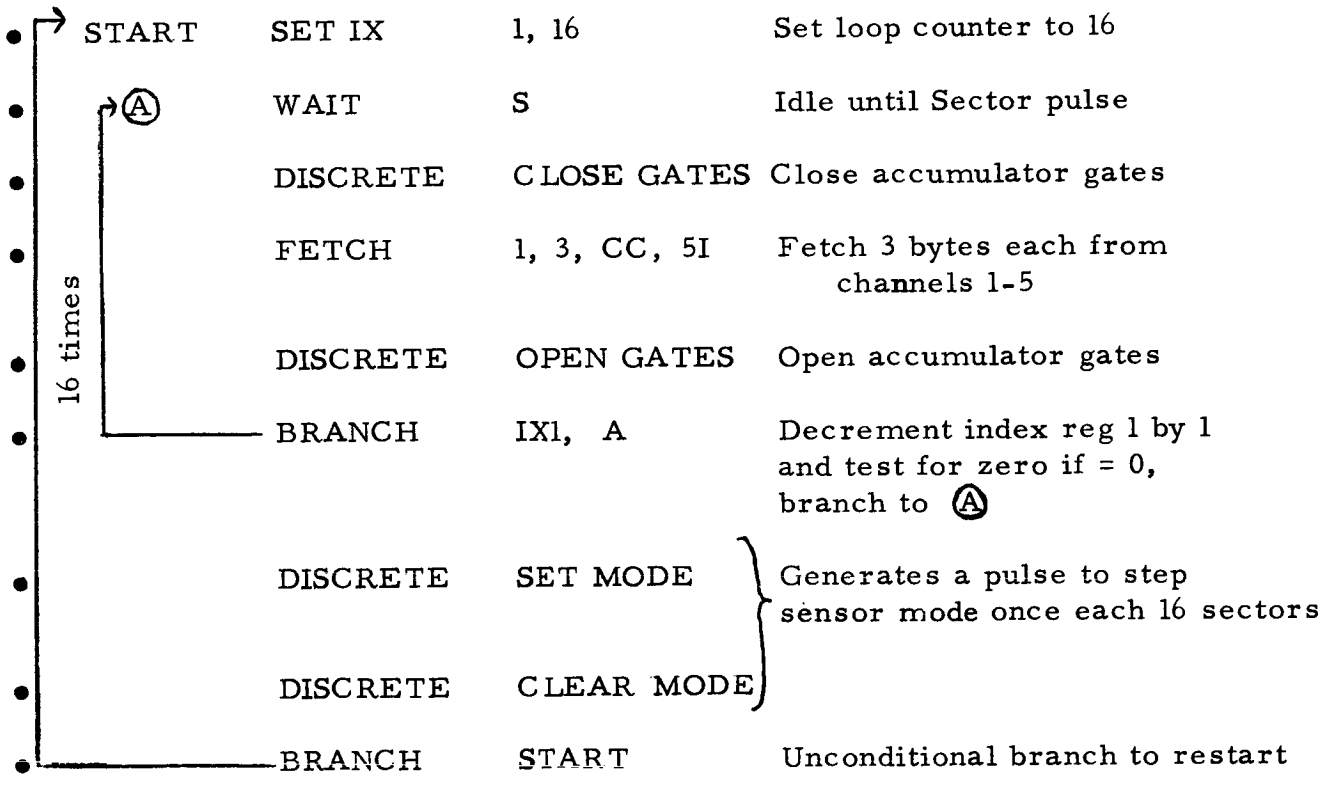
This will cause the machine to idle until a Sector pulse is sensed and then the next sequential instruction will be executed. We now have:

- WAIT S
- DISCRETE CLOSE GATES
- FETCH 1, 3, CC, 5I
- DISCRETE OPEN GATES

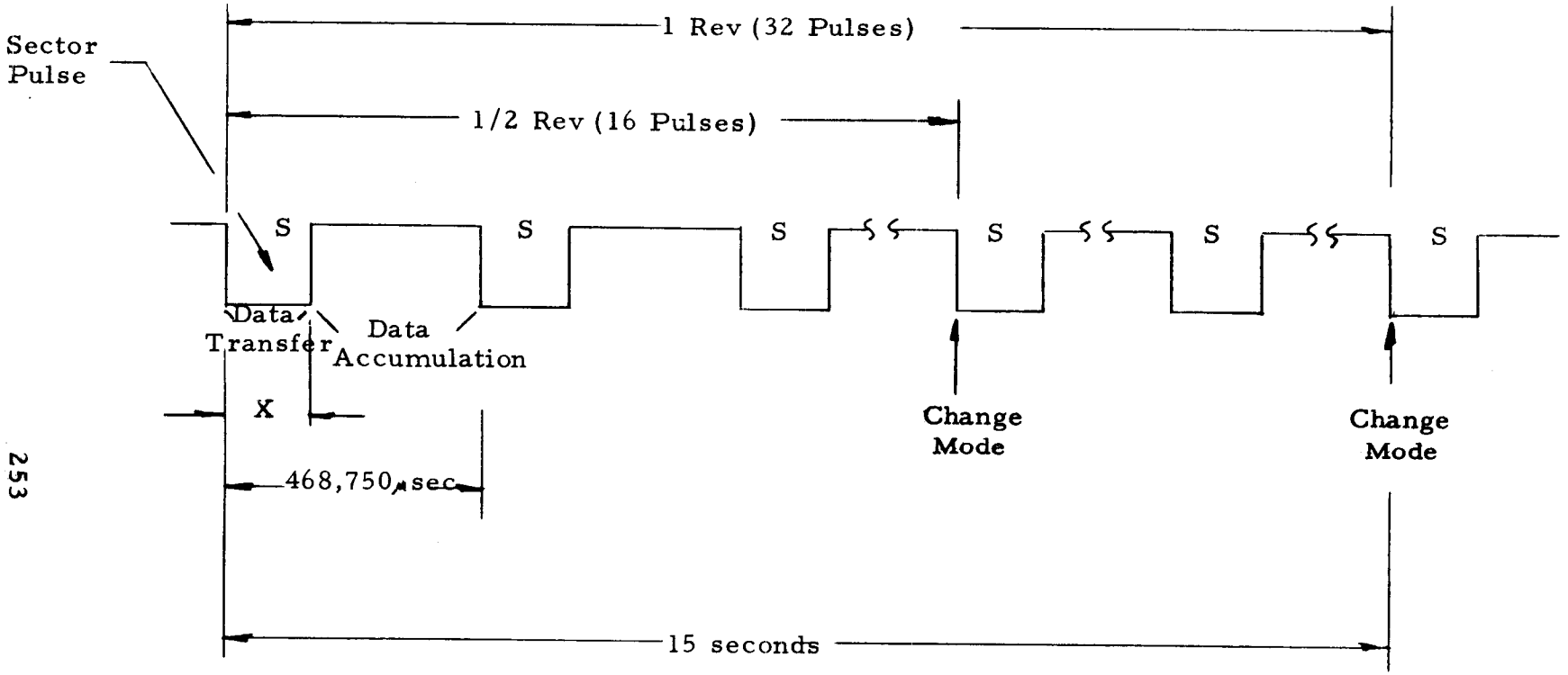
The above routine however, will obtain only one sampling of the 5 data channels and they must be sampled 32 times each revolution. In addition, the modes must be changed each 1/2 rev. or every 16 sectors (32 sectors/rev.). Therefore a loop counting technique is necessary. This is done by the use of an index register. Preceding the fetching routine, index register 1 should be set to 16 and then each time the routine is executed, decrement index register by 1. When index register 1 reaches zero, the loop will have been executed 16 times and changing the modes is necessary. The changing of the modes can be accomplished by a DISCRETE SET MODES and a DISCRETE CLEAR MODES which will generate a pulse to the stepping

logic. If the stepping logic were to automatically reset the mode discrete after stepping, the second DISCRETE instruction would not be necessary.

The program routine now becomes:



The unconditional branch to START can be a conditional branch for controlling transfer to a similar type routine which fetches data from the five channels at a different rate if desired. The above routine acquires 3 bytes each from five channels 16 times per half revolution, stepping modes each half revolution, and acquires this data continuously. In three revolutions, for example, it acquires 1440 bytes of data. The pertinent timing of the sampling is shown in Figure 3.9-1.



X = 360 to 445 μ sec depending on accumulator values; could be lengthened by 10 μ sec if output data is accessed during instruction fetch portion or data input portion of this time.

Accumulate time is $> 99.9\% \pm \approx 0.01\%$ of sector.

Figure 3.9-1 Accumulation Timing

3.9.1.2 Acquiring Analog Data

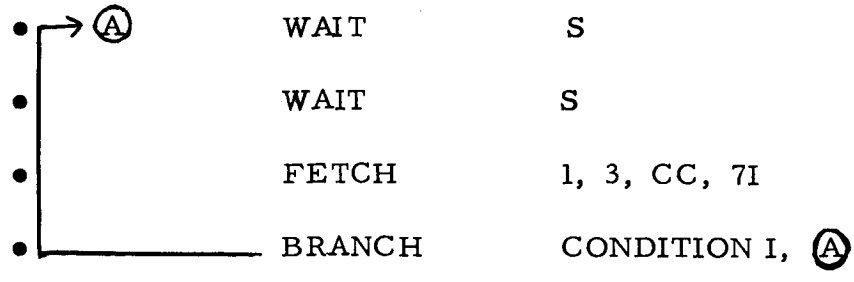
Problem: Seven (7) analog channels are to be sampled once per second with a sample rate tolerance of ± 20%, however once the rate is established it must be known to a ± 1% tolerance. Each channel contains 3 bytes of data and at certain times only 2 byte samples are necessary. Again the satellite spin rate is 4 RPM.

Solution: The data in this illustration can be fetched by a number of different methods. One method is to use a single fetch in the following format:

- FETCH 1, 3, CC, 7I

This will fetch 3 bytes from each of 7 channels starting with channel 1 and the CC indicates chain command or burst mode.

Timing control is necessary since the sampling is to be done at a 1/second rate. As in the previous illustration, a sector pulse signal will be used (32 pulses per revolution). The sector pulses occur at a 0.469 seconds (15/32) rate. Therefore, the data can be fetched in a burst every other Sector pulse or at a 0.938 second rate (within the 1/second ± 20% required). The timing can be provided by two consecutive WAIT instructions followed by the fetch.



The first WAIT results in no data being transferred and no stepping of the buffer addressing, only idle machine operation until the Sector pulse is sensed at which time the next instruction is executed. The second WAIT again causes a machine idle until another Sector pulse occurs at which time the fetch is executed. In this manner, the data is fetched every other sector. If half rate pulses were made available for wait conditions then only one wait, WAIT S/2, would be running. The routine could be followed by a branch on a condition from ground control or internal control such that the routine will be repeated until directed or programmed otherwise.

Another method for obtaining the data is to do it sequentially on pulses occuring at 8 times the frequency of the sector pulses (8S); however, this would take 8 WAIT and 7 Fetch instructions. In this case the Fetch instructions would specify command chain to cause execution on the 8S clocks. If this clock were used as the simple stepping clock available for non-chained execution, the 7 WAIT instructions could be eliminated and the 7 Fetch instructions would not specify command chaining.

When the condition specified in the BRANCH CONDITION A instruction is such that the loop is not repeated, a routine similar to that described can be executed to fetch only 2 byte samples. This routine would consist of:

• Ⓑ	WAIT	S
•	WAIT	S
•	FETCH	1, 2, CC, 7I

- BRANCH CONDITION 2, B
- BRANCH A

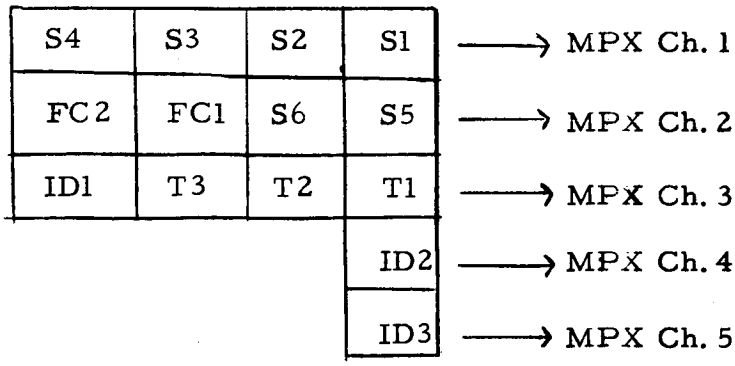
In this manner, the data fetching can be 2 or 3 bytes depending on the status of CONDITION 1 and 2 indicators. These conditions could be index registers controlling the number of times each routine (3 byte and 2 byte) is to be executed, could be conditions set by programmed discretes for alternating, or could be a number of other user specified or programmed conditions.

Because of slow rollrate variations, the actual sampling rate will vary slowly — as during the decay of the roll rate. The actual sampling rate can be determined, if important, by the relative time counts in successive frame message headers. As the counter is incremented at 0.005 second rate, time can be determined to within 0.01 seconds from one frame to the next, and more precisely over several frames. At a sample rate of 7 each second, time is known to within 1% on a frame to frame basis.

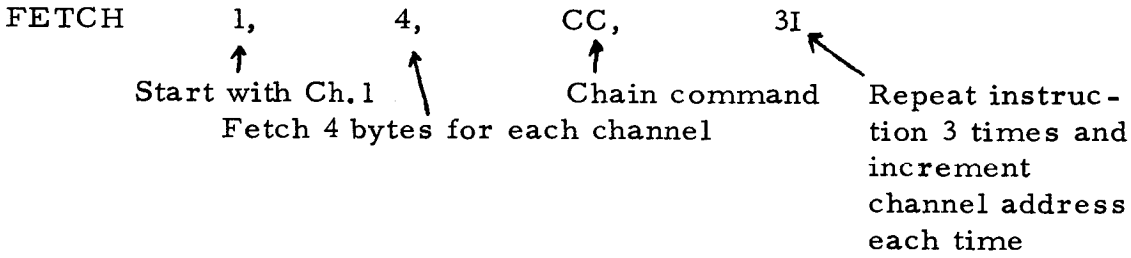
3.9.1.3 Acquiring the Message Header

Problem: The data is formated such that preceding a specified number of bytes called a frame, identification information is necessary. This information is to consist of 6 bytes of synchronization, 2 bytes containing the frame count, 3 bytes of time and 1 byte containing the frame ID code for a total of 12 bytes.

This message header data is obtained through channels with the following functional format:



Solution: To fetch the information with a frame ID of 1 involves only 1 fetch instruction with the following code:



The above instruction will fetch the 12 message header bytes (4 bytes from 3 channels) and transfer them to the buffer.

To obtain a frame ID of 3 involves the following sequence:

FETCH 1, 4, CC, 2I

Fetches 4 bytes from channels 1 and 2 (Sync and frame count)

FETCH 3, 3, CC

Fetches only 3 bytes of channel 3 data (time)

FETCH 5, 1, CC

Fetches 1 byte from channel 5

A similar routine, the last fetch being

FETCH 4, 1, CC

will provide a frame ID of 2.

The desired message header routine inserted into a program at the proper point will provide the information necessary to identify the content, format, and acquisition time of the data following the header.

3.9.1.4 Program Linkage

Problem: Assume 3 subroutines must be executed in a certain sequence other than straight sequential. These subroutines are labeled A, B, C and must be executed in order A, C, B, C, A, C, B, C after which a transfer to another section of the program, labeled D, is assumed.

Solution: The solution to the above problem could be done by repeating the subroutines in memory in the sequence stated. However, this is unduly costly from a memory usage standpoint. The logical solution is to provide linkage and control to execute the individual subroutines in the proper sequence, which means the subroutines are in memory at only one location.

The linkage and control in S^3 can be done by use of the discretes and index registers.

The flow diagram in Figure 3.9-2 represents the programming solution to illustration 4.

The programming represented by the flow diagram consists of the following:

START	SET IX	1, 2	Sets loop counter (index reg. 1) to 2
Ⓐ	RESET	DISCRETE 1	Turns discrete 1 off
	Execute	SUBROUTINE A	

Ⓒ	Execute	SUBROUTINE C	
	BRANCH	DISCRETE 1, Ⓓ	Transfers to D if discrete 1 is on
	SET	DISCRETE 1	Turns discrete 1 on
	Execute	SUBROUTINE B	
	BRANCH	Ⓒ	Unconditional branch to C
Ⓓ	BRANCH	IX1, Ⓐ	Decrements IX reg 1 by 1 and tests for zero. If not zero, branch is to A , however when zero next stored instruction is executed
	BRANCH	SUBROUTINE D	Unconditional branch to SUBROUTINE D. Not necessary if SUBROUTINE D is physically located at this point in program.

It should be noted that the BRANCH IX1 instruction performs two functions shown on the flow diagram, that is, it decrements the index register and tests for zero.

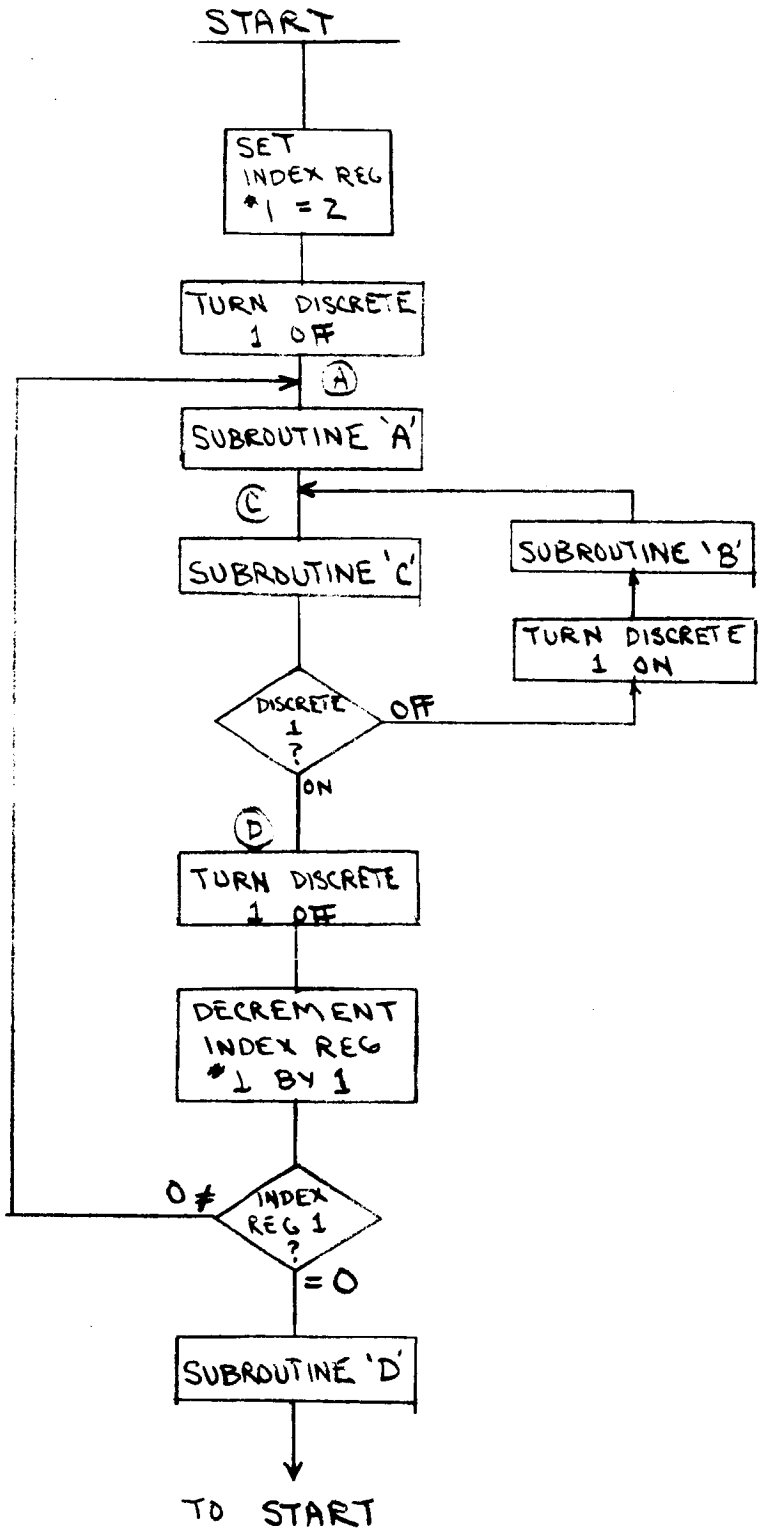
3.9.2 A Combined Program Example

This subsection presents an example of a program which collects 1 frame (256 bytes) of data, then collects another frame of data in a completely different format, repeats the process the number of times specified by the setting of index register 1 (3 times in example), then restarts the program or branches to a different program as dictated by a discrete signal controlled from the ground. During the data collection, the tape dump signal is monitored and if found off, the collected data is routed to the storage tape for later transmission to the ground receiving stations. However, if the tape dump signal is found on, the routing is directly to the transmitter for real time transmission.

BY_____DATE_____ SUBJECT ILLUSTRATION 4 SHEET NO._____OF_____

CHKD. BY_____DATE_____ FLOW DIAGRAM JOB NO._____

3.9-2



It has been assumed the data is to be collected in 6 frame blocks until the Mode signal indicates a change and that the 6 frames are to have the following format:

Frame Number	1	2	3	4	5	6
Data Types	A (32 bytes)	C (26)	A	C	A	C
	B (20)	A (32)	B	A	B	A
	A (32)	C (26)	A	C	A	C
	C (26)	A (32)	C	A	C	A
	A (32)	B (20)	A	B	A	B
	C (26)	A (32)	C	A	C	A
	A (32)	C (26)	A	C	A	C
	B (20)	A (32)	B	A	B	A
	A (32)	C (26)	A	C	A	C
	252 bytes	252				

The "A" type data consists of 4 bytes from 8 channels, the "B" type, 4 bytes from 5 channels and the "C" type, 4 bytes from 5 channels plus 2 bytes from 3 other channels. Each frame includes 3 synchronization bytes and 1 frame counter byte as identification information.

It can be seen from the frame formats that certain sequences are repeated, such as A, B, A and C, A, C. Therefore, a program with these sequences as sub-programs or subroutines is illustrated in the example to help show various program capabilities with the given instruction set. Other program solutions to the example are possible.

The subroutine program approach as coded takes 30 instructions and uses 117 bytes of memory. It collects 1512 bytes of data each complete pass. A simple program approach whereby each frame of data is collected not using the subroutine technique takes 39 instructions and 153 bytes to collect the same data.

This indicates that final program solution to the various experiment tasks depends upon a trade-off, determined by the individual mission requirements and with the S^3 system as designed plus the software versatility, the program can be changed or modified with minimum effort.

Figure 3.9-3 represents a flow diagram of the program example with the control discrete sequences on the left and the frame format in the upper right.

Program Description

This section provides a block by block description of the program flow diagram (Figure 3.9-3) for one complete program pass.

The first block from the START will initialize index reg 1 to 3. This sets up conditions for 3 passes through the major loop. Next SYNC information is fetched and stored in the leading bytes of the buffer frame. This is used for identification of the frame. Then the TAPE DUMP signal is interrogated. If found off, routing of the data is Multiplexor-to-buffer-to-tape for storage. If on, the data is routed to the transmitter instead of the tape. This allows transmission of the data while the tape is being read out, therefore data collection is not suspended during this period. Next discrete 1 is checked out and will be off at this time which causes a branch to SUBR 1.

At SUBR 1, 84 bytes of data are collected (32 "A" type + 20 "B" type + 32 "A" type) into buffer memory. Then discrete 1 is checked, found off and transfer is to SUBR 2.

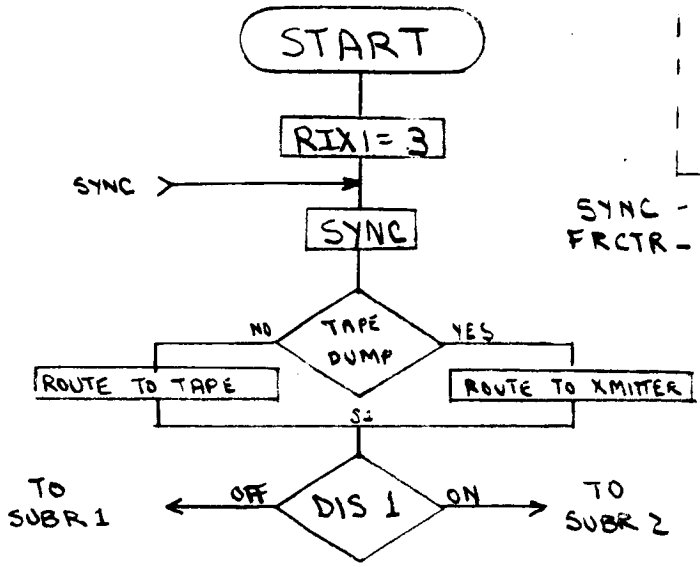
BY_____DATE_____ SUBJECT_____ SHEET NO._____OF_____

CHKD. BY_____DATE_____ PROGRAM EXAMPLE JOB NO._____

FIGURE 3.9-3 FLOW DIAGRAM

SEQUENCE

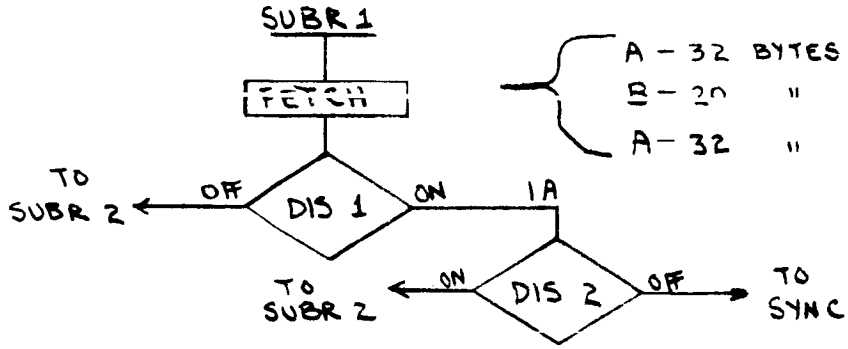
DISCRETE 1	DISCRETE 2	FETCH
OFF	OFF	SYNC 1
		2
ON	OFF	1
		SYNC 2
ON	ON	1
		2
OFF	OFF	REPEAT FROM BEGINNING

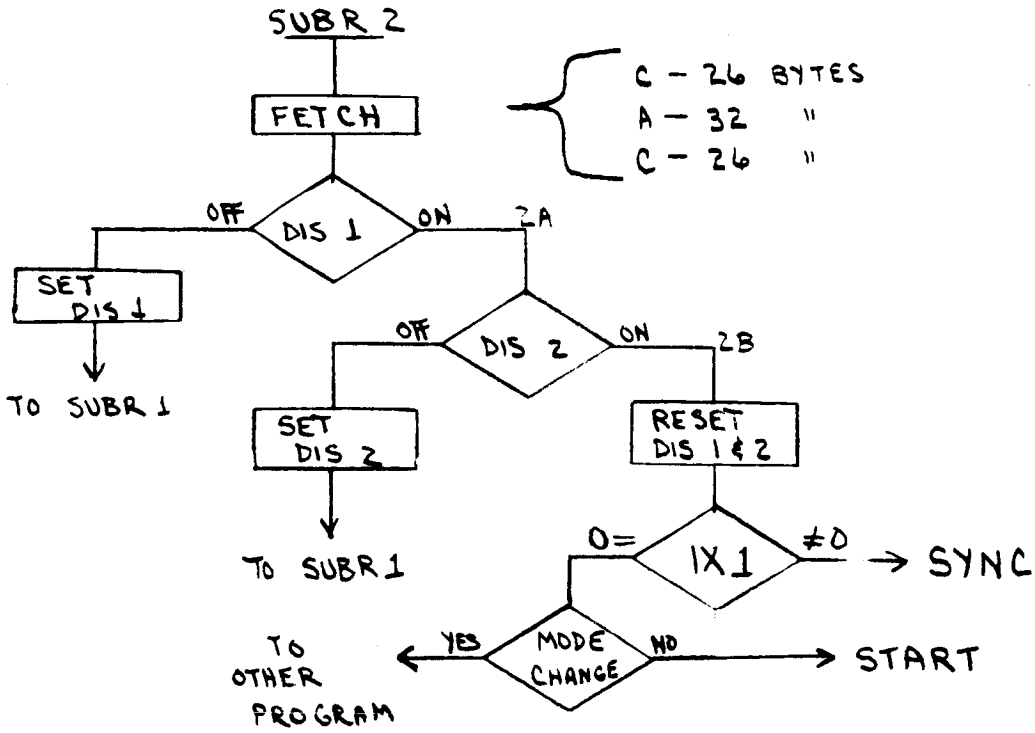


FRAME FORMAT

FRAME →	1	2	3	4	5	6
SUBR	1	2	1	2	1	2
	2	1	2	1	2	1
	1	2	1	2	1	2

SYNC - 3 BYTES
FRCTR - 1 BYTE





SUBR 2 will collect 84 bytes of data (26 "C" type + 32 "A" type + 26 "C" type) and then check discrete 1 which is still off. Discrete 1 is then set and branch is to SUBR 1. SUBR 1 fetches another 84 bytes of data and discrete 1 is now on, therefore branch is to 1A. Then check discrete 2 which is off and branch to SYNC. Up to this point 256 bytes have been loaded into memory (4 SYNC + 3 x 84 data) which completes frame 1.

SYNC fetches 4 bytes to start frame 2 and checks the tape signal as described before. Discrete 1 is found on this time and the branch is to SUBR 2.

SUBR 2 fetches 84 bytes of data, check discrete 1 which is on, checks discrete 2 which is off and then is set on and branches to SUBR 1.

SUBR 1 again fetches 84 bytes, checks discretes 1 and 2 which are both on and then branches to SUBR 2.

SUBR 2 fetches 84 bytes which completes frame 2. Then check discrete 1 and 2, both on, therefore at 28 discretes 1 and 2 are reset or turned off. This is back to their initial state. Then index reg 1 is decremented by 1, tested for zero and if not zero branches to SYNC which will cause 2 more frames to be generated.

This process will continue for 6 frames of data and then index reg 1 will be zero and branch to where the ground controlled mode signal is strobed. If no mode change is indicated the program is repeated and 6 more frames of data will be collected. If a mode change is detected the program designated will now be executed.

	Instruction	Inst. Length (Bytes)	Comments	Data Bytes Fetched
START	ROUTE M-B, B-T, C, 0, 5	3	Clears address registers and sets buffer frames as 0 thru 5.	
	SET IX 1, 3	4	Set index reg 1 to 3 which will cause 6 frames to be collected.	
SYNC	FETCH SYNC, 4B, CC, 3I	4	Fetches Sync, FRCTR, time, and ID	12
	BRANCH TAPE DUMP, DUMP	4	Checks tape dump signal, if on branches to DUMP	
	ROUTE M-B, B-T	3	and if off routes to tape for storage.	
S1	BRANCH DIS 1, SUBR 2	4	Check discrete 1, will alternately be OFF & ON	
SUBR 1	FETCH A1, 4B, CC, 8I	4	Fetches 4 bytes from 8 channels starting with channel A1.	4 x 8 = 32
	FETCH B1, 4B, CC, 5I	4	Fetches 4 bytes from 5 channels starting with channel B1.	4 x 5 = 20
	FETCH A1, 4B, CC, 8I	4		4 x 8 = 32
	BRANCH DIS 1, 1A	4		
	BRANCH SUBR 2	4	Unconditional branch to SUBR 2	
1A	BRANCH DIS 2, SUBR 2	4		
	BRANCH SYNC	4		
SUBR 2	FETCH C1, 4B, CC, 5I	4	Fetches 4 bytes from channels C1-C5.	4 x 5 = 20
	FETCH C6, 2B, CC, 3I	4	Fetches 2 bytes from channels C6-C8.	2 x 3 = 6
	FETCH A1, 4B, CC, 8I	4		4 x 8 = 32
	FETCH C1, 4B, CC, 5I	4		4 x 5 = 20
	FETCH C6, 2B, CC, 3I	4		2 x 3 = 6
	BRANCH DIS 1, 2A	4		
	SET DIS 1	4	Linkage control	
	BRANCH SUBR 1	4		
2A	BRANCH DIS 2, 2B	4		
	SET DIS 2	4	Linkage control	
	BRANCH SUBR 1	4		
2B	RESET DIS 1 & 2	4	Turns discrete 1 & 2 off to initial state	
	BRANCH IX1, SYNC	4	Decrements index reg 1 by 1 and tests for zero	
	BRANCH MODE, NEW	4	Checks Mode control, if changed branch to program NEW.	
	BRANCH START	4	Branch to START and repeat program if no mode change in indicated.	
DUMP	ROUTE M-B, B-X	3	If tape dump signal is on this will be executed	
	BRANCH S1	4	& route data to xmitter for real time transmission.	

*NEW is start of other program routine dependent upon requirements of experiments.

30 Instructions 117 Bytes

3.10 S^3A Sizing

The S^3A experiment was selected for sizing as it is representative of the worst case experiment. The experiment was programmed to verify the adequacy of the S^3 instruction set and to determine the memory requirements for instructions and data buffer.

The coded program and buffer memory can be contained well within a 4K memory, with considerable memory space available for program expansion or modification.

(In the following, the term "filters" is used to denote different modes of operating experiment sensors. These modes might involve stepping of switches, changing voltages, incrementing wheels or other such action to alter the performance characterisitcs of the sensors.)

The S^3A experiment is functionally represented in Figure 3.10-1 and has the following data sampling requirements:

- Particles (A type data in this report)

  Collect 9 bits (3 bytes) from each of 5 channels for each sector (32 times/rev), except for 10% of the time during which collect 2 bytes from 2 channels at a 16/sector rate. Change filters each 16 sectors and reset filters at 3 rev intervals. Accumulators are to be operated simultaneously and accumulate for at least 80% of each sector with a variation of no more than $\pm 1\%$ of a sector.

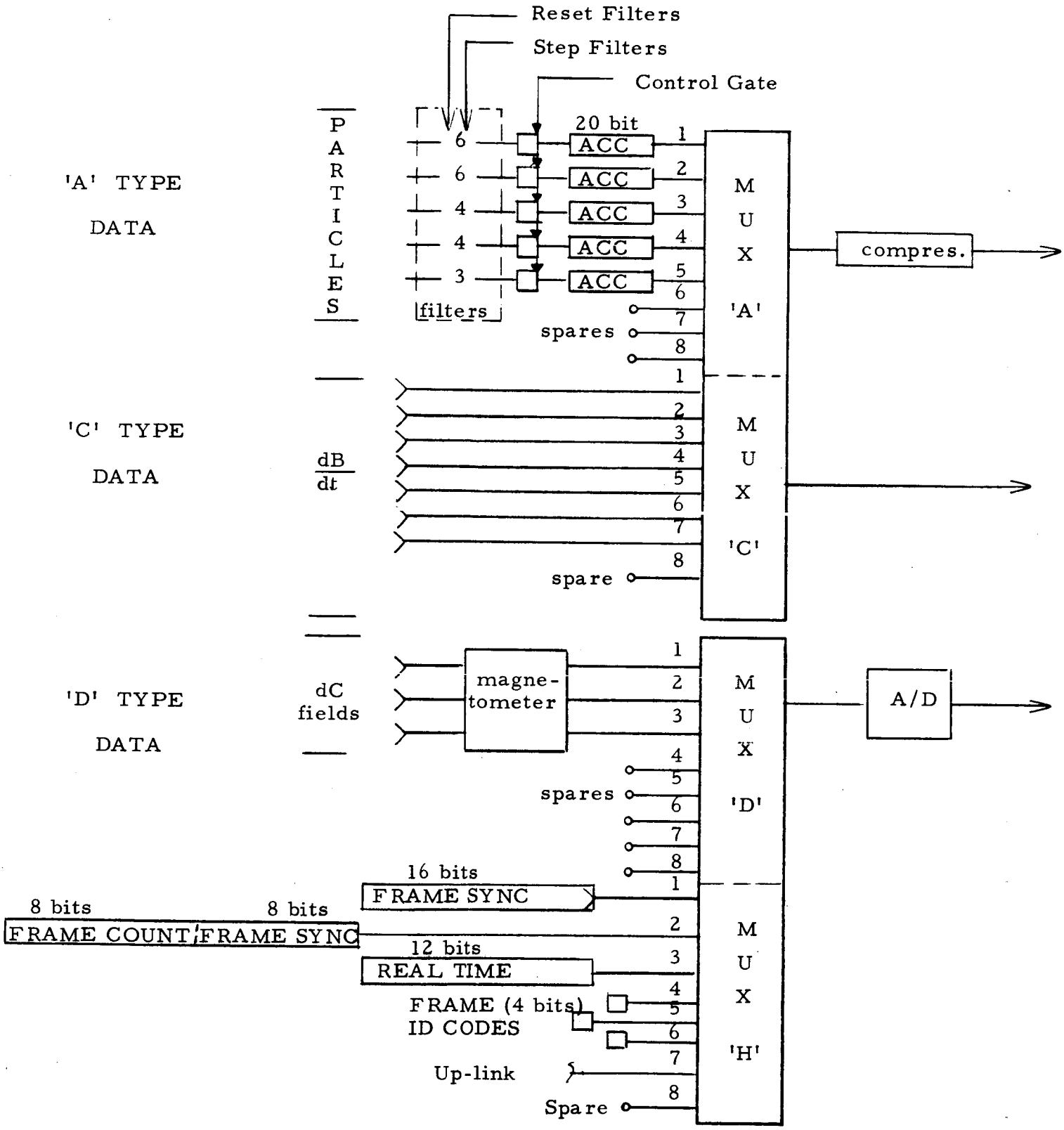


Figure 3.10-1 S/A Functional Interfaces

- DB/DT (C type data)

  Collect 6 bits (2 bytes) from each of 7 channels 1/second. Rate can be ±20% and set or determinable to ±1%.

- D-C Fields (D type data)

  Collect 10 bits (3 bytes) from 3 channels 3 times per second except for 10% of the time during which collect 2 bytes at a 30 times per second rate. Rates can be ±20% and set or determinable at ±1%.

- Housekeeping

  In addition to experiment data, certain vehicle housekeeping data must be gathered by the program at regular intervals.

The experiments, from a programming standpoint, involve collecting the various types of data at the specified rates and storing the data in a known format into a buffer memory. The data will then be transferred onto the storage tape. In addition, other functions must be controlled at specified intervals such as changing the particle sensor modes (or filters).

The process repeats until the satellite is within contact of a ground receiving station at which time the information stored on the tape is transmitted to the receiving station. Data collection is to continue during the tape dumping process, therefore, the transfer of data from the buffer memory to the tape must be terminated and transfer now is from memory to the transmitter for real time transmission of the data collection.

Several approaches to the programming solution of the experiment were investigated. In evaluating these various program approaches, several factors enter into the final solution. A few of the factors are: Buffer memory size, Program instruction count, Tape speed and best utilization of the tape storage.

- The buffer memory size is of prime importance since this is the major contributor in determining the final overall memory size.
- The program instruction count must be taken into account also. However, in the programs investigated for this particular experiment, the buffer memory was significantly larger than any of the program instruction counts (bytes).
- The tape speed, being fixed and not synchronized with the collection of the experiment data, has a direct bearing on the buffer memory size. In addition, the tape speed also affects the density of the information stored.
- Utilization of the tape storage can be a major factor since the data being collected must be stored on the tape until a ground station can receive the tape readout. Of necessity, some of the data going to the tape may not be message header, experiment, or vehicle data but will be spare bytes as necessary to round out frames to 256 bytes. The number of spares will be a function of the quantity and repeatability of the data sampled and will be a function of the program

approach. To reduce the number of spares requires more instructions in the program and requires more unique frame types to be used.

3.10.1 Single Program Approach

In this approach, a single program is used to collect all data and the sample rate is derived from the DSC. Should the DSC fail, the same program could be driven by the SCC.

Simple Program

The easiest program approach is to assign each sector as a frame and repeatedly fetch data, a sector at a time, with proper Sync information into the buffer memory, from which transfer is made automatically to the tape. This type program requires the least programming know-how, few instructions and is the least complex. Total instruction count was estimated at 125 taking a total storage of 370 bytes.

However, in the experiment defined, the amount of data collected during a sector is not 256 bytes (a full frame) and therefore spare bytes are required to round out each frame or sector of data. These spares naturally get loaded onto the storage tape along with the experiment data, and take up specific slots in the format. Using this scheme, the spares occupy approximately 24% of the storage tape. In cases where the tape storage available is not fully utilized or time between ground receiving stations is short, the simple program approach may be adequate.

Data Compaction - Straight Line Approach

The obvious answer to the disadvantages of the Simple Approach is to compact the experiment data such that less spare bytes are necessary to complete a frame of memory. This implies that more than a sector of data is contained in each frame of memory and this does present certain control problems such as generating the Sync information at the proper sequence in a loop type program. For this reason, a straight line approach was investigated. That is to say, the program is executed in sequential, non-looping order. This overcomes the disadvantage of the tape utilization, however, the instruction count (bytes) becomes large in comparison to the Simple program. This type program is very straight forward, easily understood and usually a starting point for more sophisticated programming. Total instructions are 380 taking a total of 1040 bytes of storage.

Data Compaction - Subroutine Approach

It was found that in the straight line approach certain functions were being done repeatedly. Therefore, these functions were programmed into Subroutines or as separate programs and these subroutines are tied into one master program by various control and loop routines.

This approach minimizes all the factors except for the instruction count and the program becomes more complex. Program complexity may affect the ease with which the program can be modified or added to. However, the memory organization and the flexible instruction set allow program changes to be made readily. The instruction storage is

approximately one-half of the previous approach and about 1 1/3 times the simple approach. The Subroutine approach was felt to be the best trade-off and it, therefore, is discussed below. Instruction count in this tape-optimized approach is 152 instructions taking a total of 555 bytes of instruction storage.

3.10.1.1 Program Description

Sampling

Based on the experiment requirements, the program is divided into two distinct sections. One section to handle the data collection during the 10% high resolution rate and the second section to handle the remaining 90% of the three rolls or revolutions of the satellite.

In 3 revolutions of the satellite, the program will execute one complete pass. Each revolution of the satellite is divided into 32 sectors and each sector is sub-divided into 32 sub-sectors identified by the SS signal to yield a basic clock rate at 4 RPM of 68 pulses per second. Since the program cycles at a 3 revolution rate, a total of 96 (3 x 32) sectors of experiment data are collected in each complete pass through the program.

The program described in this section provides the following data sampling rates:

- Particles — Collects 3 bytes from each of 5 channels in sector (32 times/rev) - Same as required by experiment.

Collects 2 bytes from 2 of the channels at a 16/sector rate for 10 of 96 sectors (10.4%) - Requirement is for only 10%.

Resets filters at 3 rev intervals and steps filters each 16 sectors.

- dB/dt

  Collects 2 bytes from each of 7 channels 1/.9375 second - requirement is 1/second (±20%).

- D-C Fields

  Collects 3 bytes from 3 channels 3 times per .9375 seconds - Requirement is 3/second. Collects 2 bytes at a 32 times per .9375 second rate - Requirement is 30 times per second - for same 10% interval as the particle data.

- Housekeeping

  Not fully defined - program setup to collect 14 bytes at 32 times per .9375 second.

Figures 3.10-2, 3.10-3, and 3.10-4 illustrate the data framing operations and the basic use of the subsector clocks.

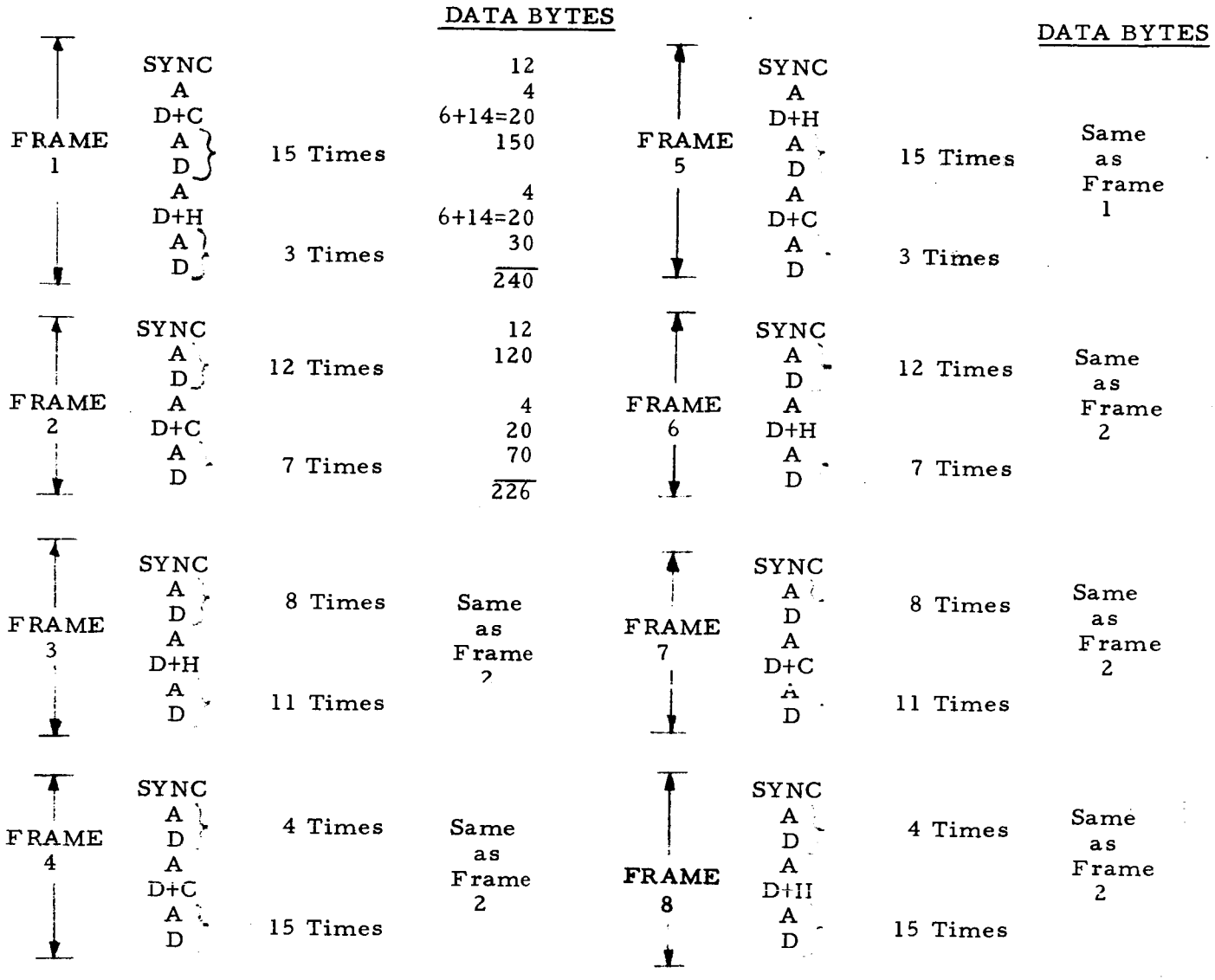


Figure 3.10-2 S^3A Frames 1 through 8, Formats

Figure 3.10-3 S^3A Data Sampling, Frames 1 through 8 (Sector 1 through 10)

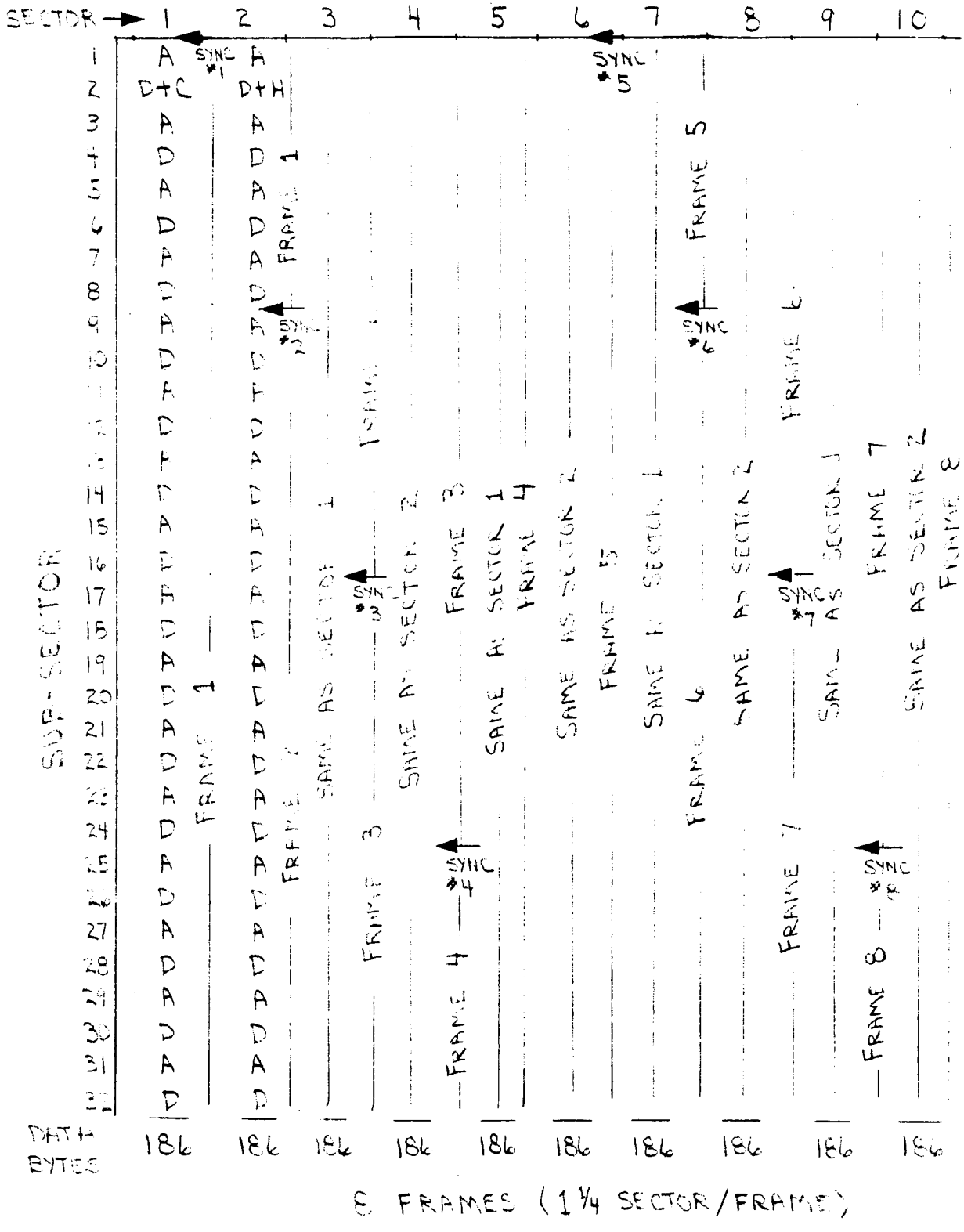
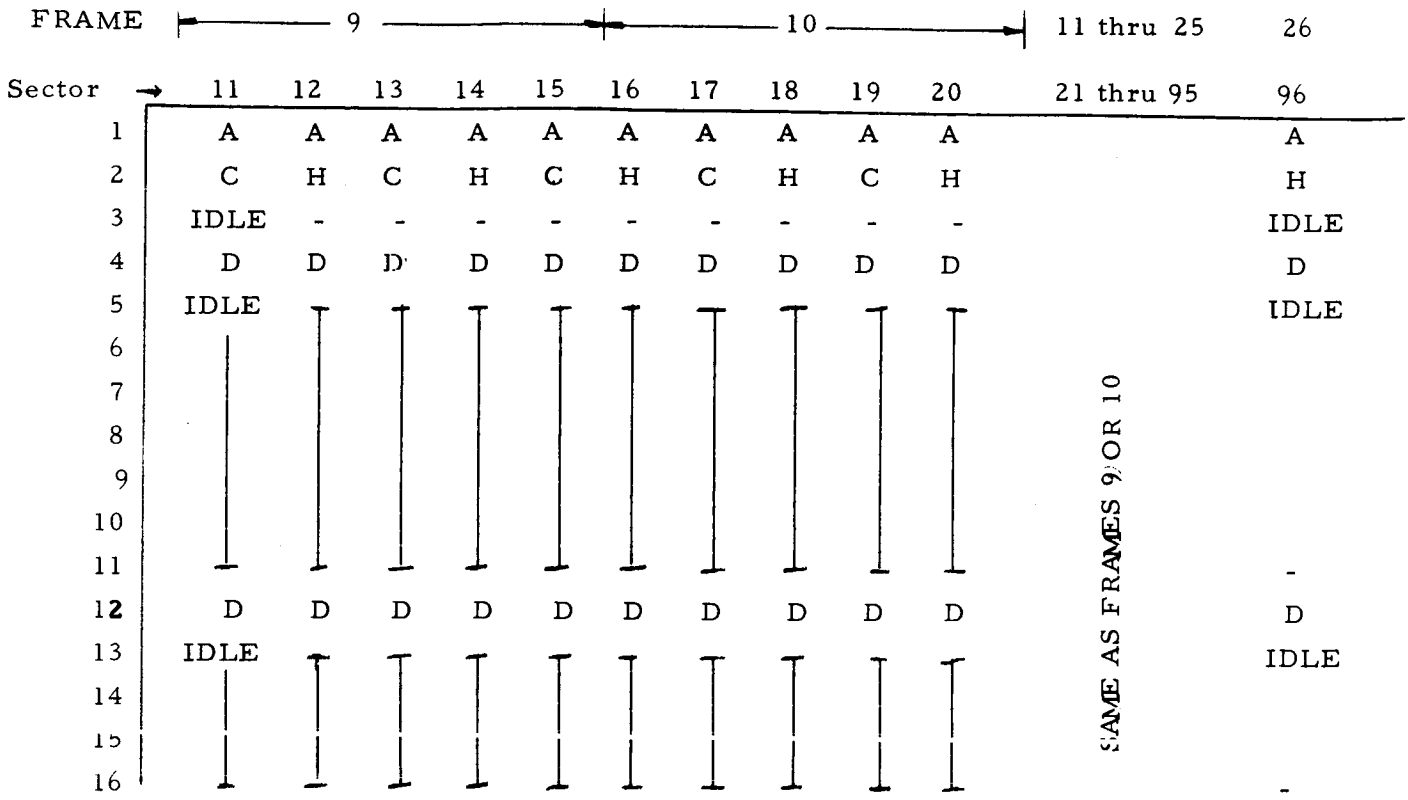
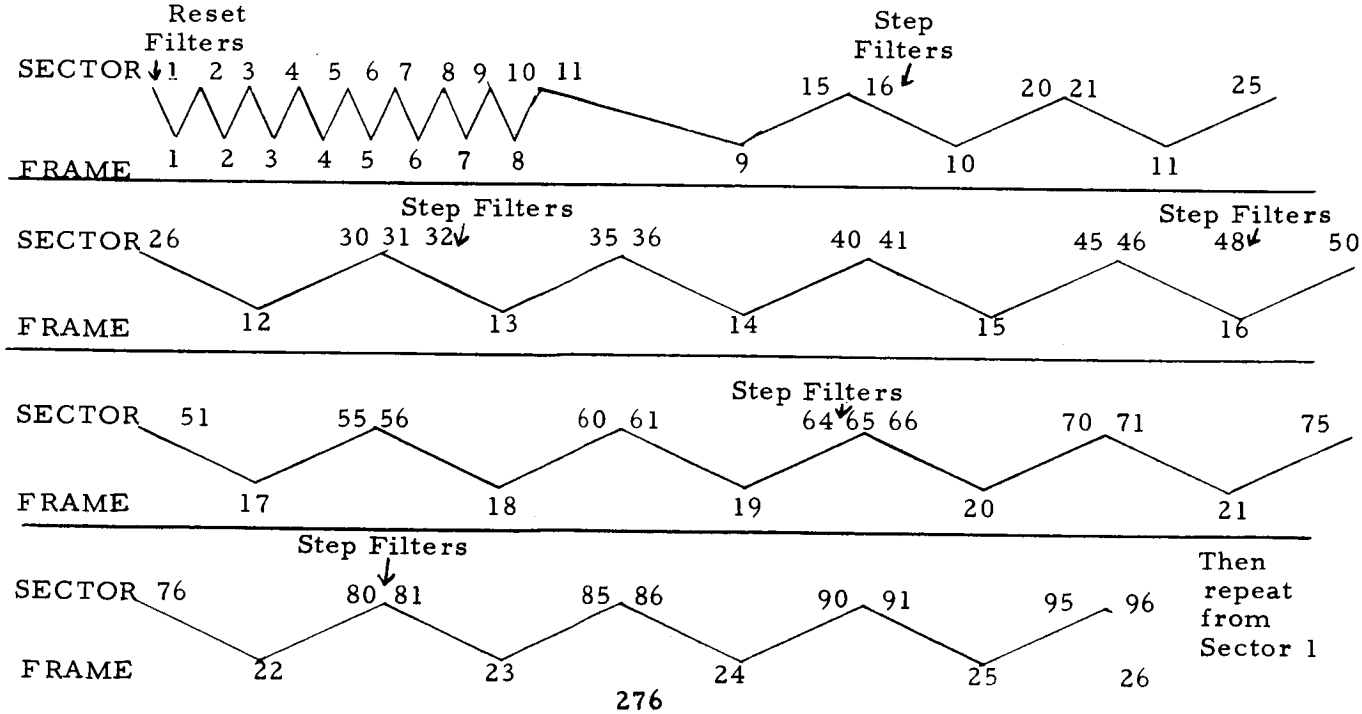


Figure 3.10-4 S^3A Data Sampling Frames 9 through 26 (Sectors 11-96)



OVERALL 3 REV FORMAT



The data sampling requirements of the experiment are met in all cases and in some cases exceeded, in the program designed, using the DSC clock.

First Section

The first section of the program must collect 10% of the 96 sectors or 10 sectors of data. The buffer storage for this data is in terms of frames (256 bytes of memory). Each frame contains Sync, F rctr (Frame Counter), Time and ID information as the first 12 bytes, the remaining 244 bytes can be used for data. Since each of the first 10 sectors contain only 174 data bytes, the frames were laid out to contain 1 1/4 sectors. Therefore, the first 10 sectors collect eight frames of data. Figures 3.10-2 and 3.10-3 show the format of frames 1 through 8. In this manner, the number of spare bytes necessary to round the frame to 256 is greatly reduced. As shown in Figure 3.10-3, the "A" type data is collected every other sub-sector (16 times/sector), the "D" type data every other sub-sector (16 times/sector) and the "C" type data once every other sector, with "H" (housekeeping) type data taking place in sectors not containing "C" data. Upon completion of the collection of the first 10 sectors of data, control is transferred to the second section of the program for collection of data sectors 11 through 96.

Second Section

The second section of the program requires sampling at a slower rate and therefore a 16 sub-sector (SS/2) scheme was employed.

Figure 3.10-4 shows the format for sectors 11-96 (Frames 9-26). Each sector of data contains 47 data bytes; therefore, 5 sectors are collected into 1 memory frame for a total of 247 bytes (12 Identification bytes + 5 x 47 data) leaving 9 spare bytes for frames 9-25. Frame 26 contains only 1 sector of data and therefore has 197 spares (256 - (47 + 12)).

Each sector (11 - 96) collects one "A" type, two "D" types and either a "C" or "H". In addition, IDLE cycles are programmed for proper timing of the data collection.

The lower half of Figure 3.10-4 shows the overall Sector and Frame relationship and points in the program where the "A" filters are changed.

Program Flow

Figures 3.10-5 and 3.10-6 (4 pages) represent the Detail Program Flow. The first 8 frames, as shown in Figure 3.10-6 can be divided into 3 distinct subroutines which are to be executed in various sequences and a number of times. These three subroutines are (1) Fetch A and D+C, (2) Fetch A and D, and (3) Fetch A and D+H. The major portion of the Program flow and instructions consists of the control logic into and out of these 3 subroutines.

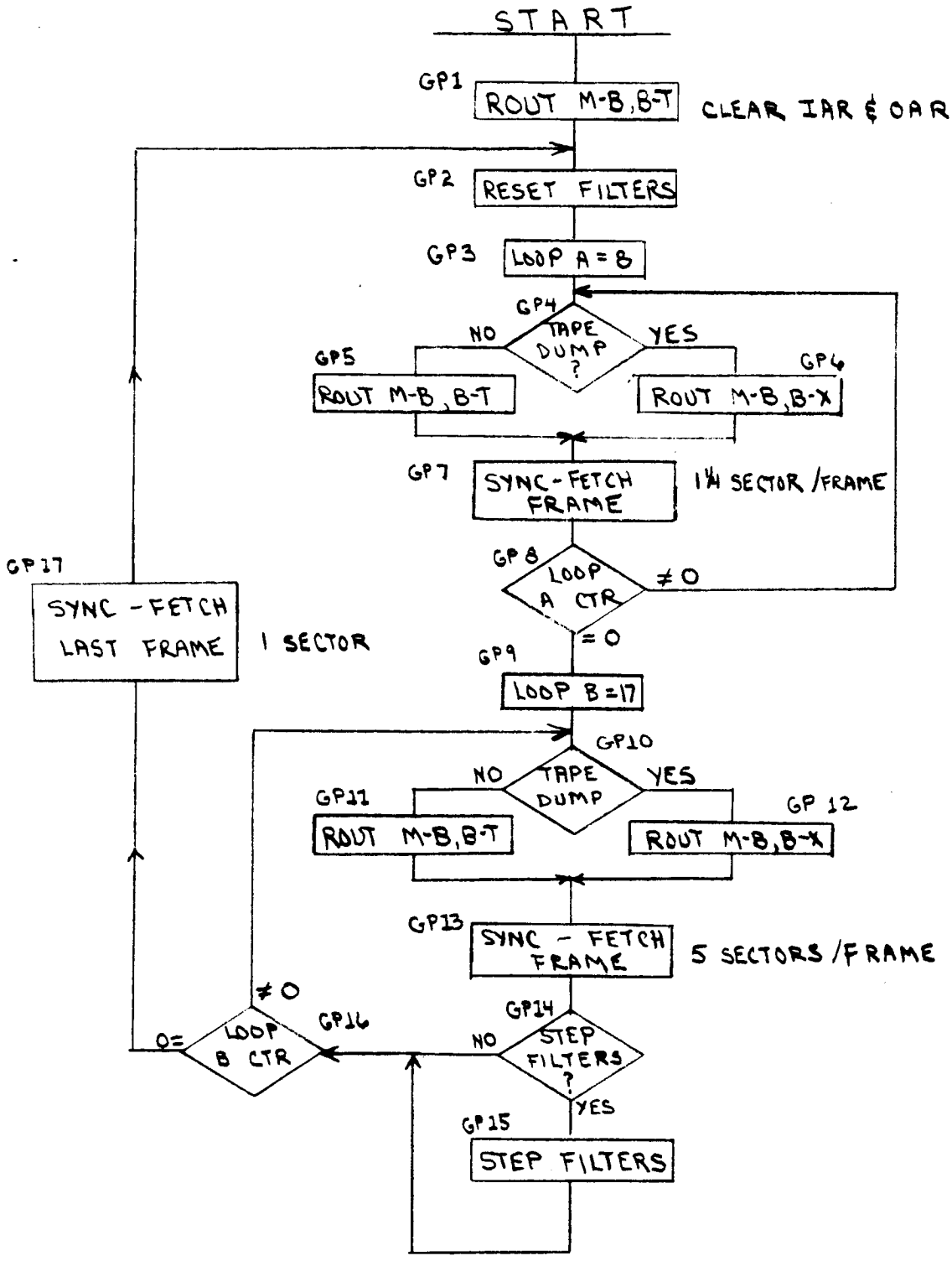
At the bottom of sheet 1 of Figure 3.10-6, the condition of the four controlling discretes are noted in sequence.

Each block is identified by a DP number.

BY_____ DATE_____ SUBJECT 5.2 SHEET NO_____ OF_____

CHKD. BY_____ DATE_____ GENERAL PROGRAM JOB NO_____

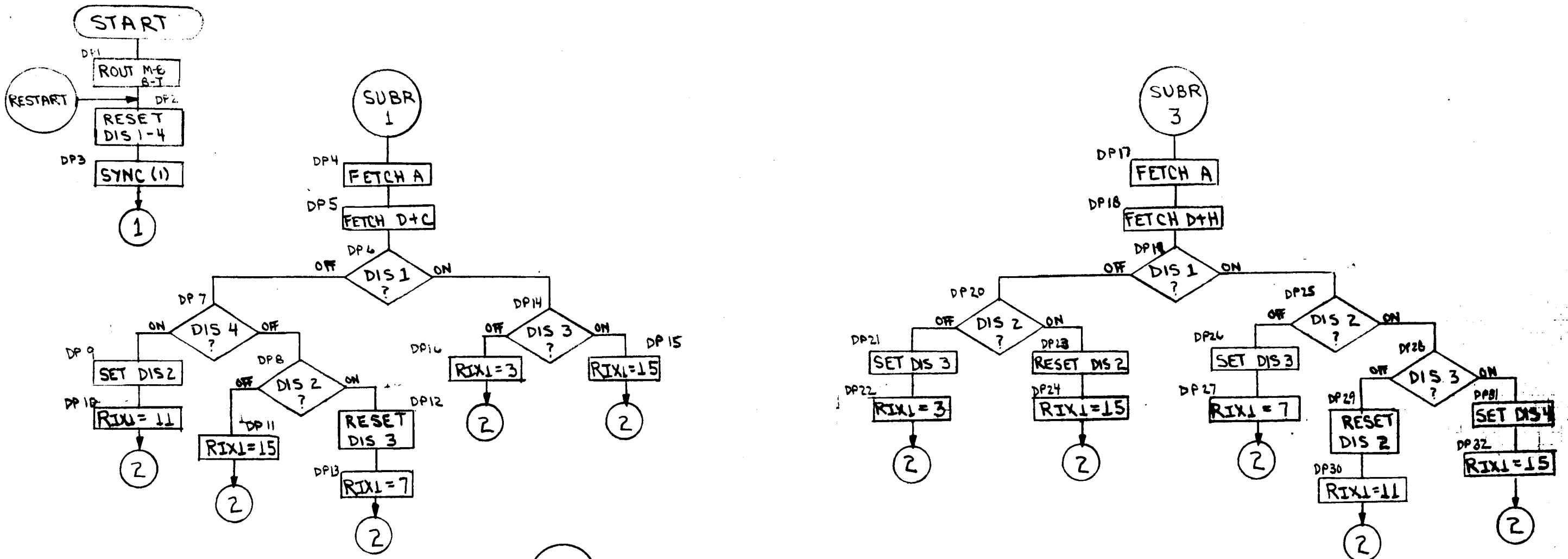
FIG. 3.10-5 FLOW (3 REV PROGRAM)



BY..... DATE..... SUBJECT S3 SHEET NO. 1 OF 3

CHKD. BY..... DATE..... DETAIL PROGRAM JOB NO.

FIGURE 3d0-6 FLOW (FRAMES 1-8)

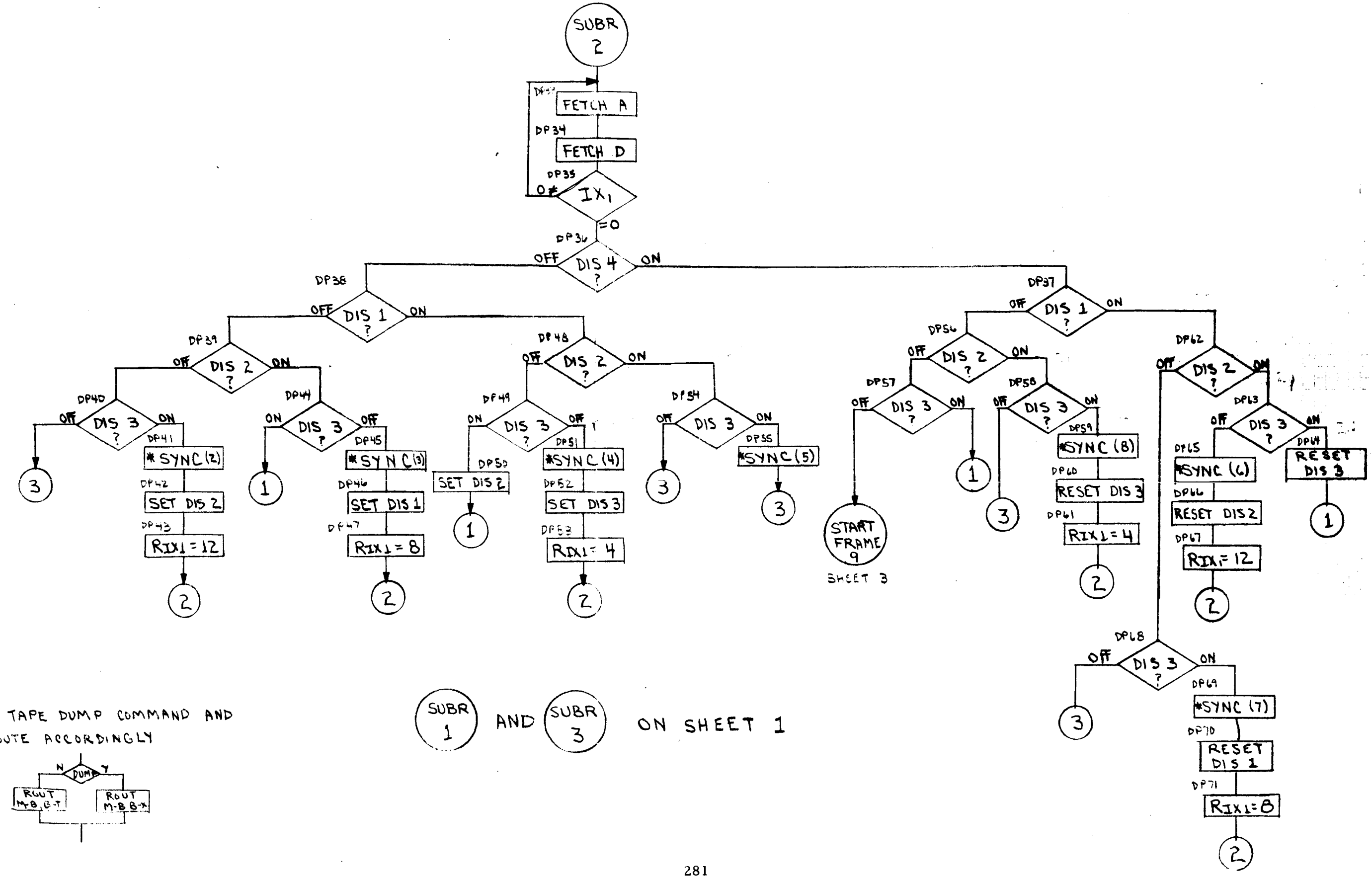


FRAME→		1	2		3		4		5			6		7		8	
DISCRETE	1	OFF	–	–	–	ON	ON	ON	ON	ON	ON	ON	ON	OFF	–	–	–
	2	OFF	–	ON	ON	ON	OFF	–	ON	ON	ON	OFF	–	–	ON	ON	OFF
	3	OFF	ON	ON	OFF	–	–	ON	ON	ON	OFF	–	ON	ON	ON	OFF	–
	4	OFF	–	–	–	–	–	–	–	ON	ON	ON	ON	ON	ON	ON	ON

BY..... DATE.....
CHKD. BY..... DATE.....
FIGURE 3.10-6

SUBJECT S³ DETAIL PROGRAM FLOW (FRAMES 1-8)

SHEET NO. 2 OF 3
JOB NO. .....



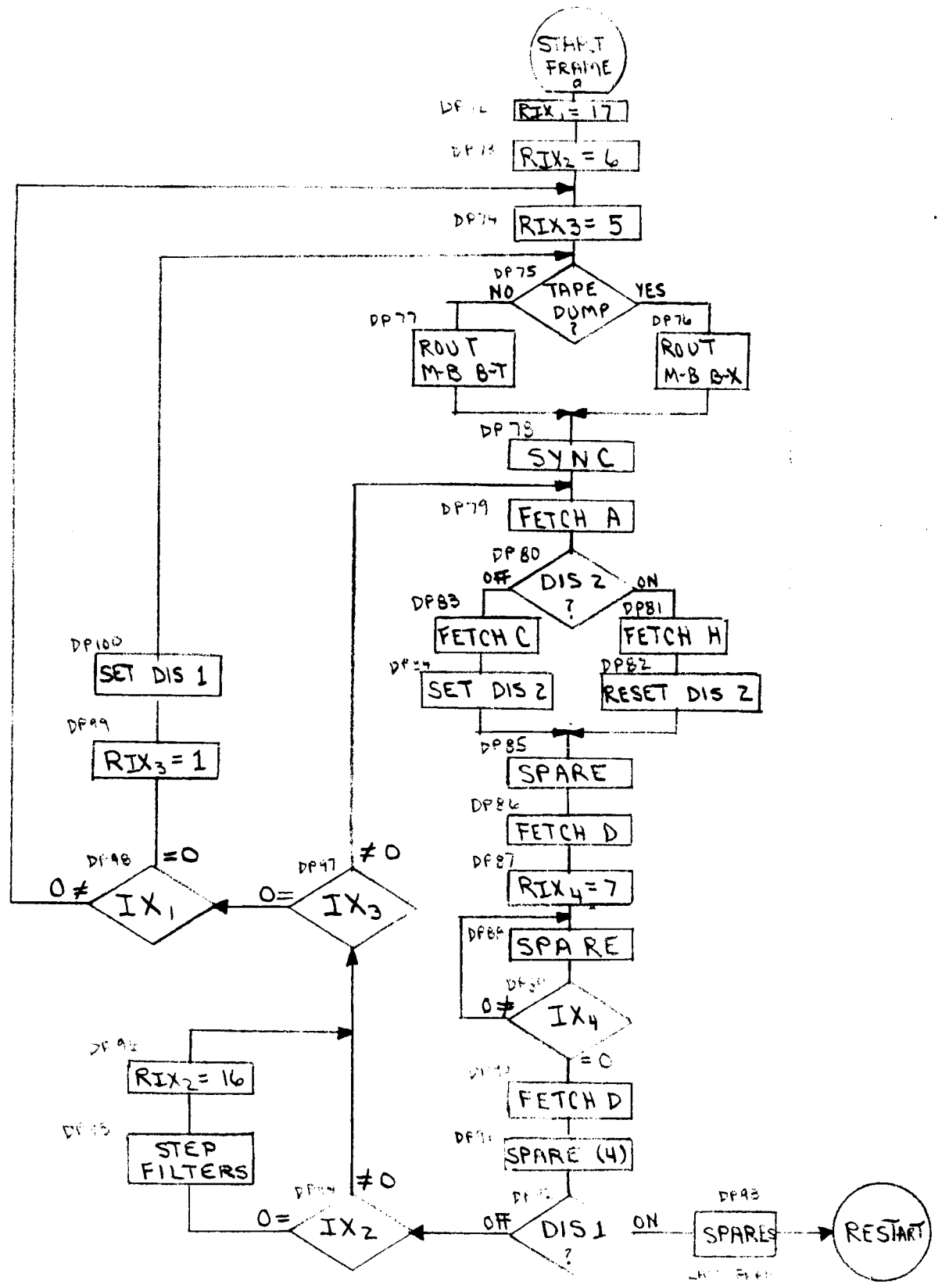
* CHECK TAPE DUMP COMMAND AND ROUTE ACCORDINGLY

SUBR 1 AND SUBR 3 ON SHEET 1

BY..... DATE.....
CHKD. BY..... DATE.....
FIGURE 3.10-6

SUBJECT S3 DETAIL PROGRAM FLOW (FRAMES 9-26)

SHEET NO. 3 OF 3
JOB NO. .....



Coded Program

The Worst Case Program coding is shown in symbolic representation. The program is represented in diagram form in Figures 3.10-5 and 3.10-6 and reference should be made to these figures in following the program coding.

Along with the coding, each instruction length in bytes is indicated and when applicable, the number of bytes of data being fetched. Comments accompany certain instructions or groups of instructions.

The coded program may not coincide with the flow diagrams in certain instances since certain programming techniques to optimize memory usage are not shown on the flows.

The following particulars may help in better understanding the program.

1. Frames 1 through 8 section
   - Data is collected on 32 sub-sector basis (SS signal)
   - Each frame contains 1 1/4 sector of data for a total of 10 sectors in the 8 frames.
   - Frames 1 and 5 contain 240 data bytes with 16 spares.
   - Frames 2-4 and 6-8 contain 226 data bytes with 30 spares.
   - The 8 frames contain 1836 bytes of data and 212 spare bytes.

2. Frames 9 through 26 section
   - Data is collected on a 16 sub-sector basis (SS/2 signal)
   - Each frame contains 5 sectors of data except the 26th frame which contains 1 sector of data for a total of 86 sectors of

data (5 x 17 + 1).

- Frames 9-25 contain 247 data bytes with 9 spares.

  Frame 26 contains 59 data bytes with 147 spares due to the nature of the problem and the way it was programmed.

  The 18 frames contain 4258 bytes of data and 35 spare bytes.

3. Complete Program

   - 3 REV (96 sector) program - generating 26 frames of data (256 bytes/frame)
   - Total instruction count is 152 using 555 bytes of memory.
   - Load program and index regs use 24 bytes.
   - Based on a 6 frame buffer the total memory usage is 2115 bytes (578/610 + 1536 + 24).

     Total bytes collected or generated by the program is 6656 (26 x 256) of which 6094 are data and 562 spares. The spares occupy 9.2% of the tape. These are available for additional experiments and/or housekeeping.

3.10.6.2 Coded Instruction - Frames 1-8

Detail Block No.	Instruction	(Bytes) Length	Comments	Data Bytes (Fetched)
	START			
DP1	ROUT M-B, B-T, C	3	Sets for loading into buffer at 0	
DP2	RESET DIS 1-4 & Filters	4	Initialize discretes 1-4 and filters	
DP3	FETCH, H1, 4B, CC, 3I	4	Fetches SYNC, FRAME COUNTER,	8
			REAL TIME and FRAME ID1	3
				1
	SUBR 1			
DP4	WAIT SS	2	Idles awaiting a Sub-Sector Pulse (SS)	
	RESET GATE A	4	Close gate to particle accumulators	
	FETCH A1, 2B, CC, 2I	4	Fetch particle data	4
	SET GATE A	4	Open Gate	
	FETCH Spare, 4B, CC, 2	4	Spares necessary to round frame to 256 bytes	8
DP5	WAIT SS	2		
	FETCH D1, 2B, CC, 3I	4	Fetch DC Field Data	6
	FETCH C1, 2B, CC, 7I	4	Fetch dB/dt data	14
DP6	BRANCH DIS 1, DP14	4	Branch to DP14 if discrete 1 is on	
DP7	BRANCH DIS 4, DP9	4		
DP8	BRANCH DIS 2, DP12	4		
DP11	SET IX 1, 15	3	Set IX reg 1 to 15 for number loops thru DP33 and DP34	
	BRANCH DP33	4	Unconditional branch to DP33 (SUBR 2)	
DP14	BRANCH DIS 3, DP15	4		
DP16	SET IX 1, 3	3		
	BRANCH DP33	4		
DP9	SET DIS 2	4	Set discrete 2 (Control logic)	
DP10	SET IX 1, 11	3		
	BRANCH DP33	4		
DP12	RESET DIS 3	4		
DP13	SET IX 1, 7	3		
	SUBR 2			
DP33	WAIT SS	2		
	RESET GATE A	4	Fetch "A" Data	
	FETCH A1, 2B, CC, 2I	4		4

DP34	WAIT SS	2	Fetch "D" data	
	FETCH D1, 2B, CC, 3I	4		6
DP35	BRANCH IX_1, DP33	4	Decrement IX reg 1 by 1 and branch to DP33 until IX reg 1 becomes zero.	
DP36	BRANCH DIS 4, DP37	4		
DP38	BRANCH DIS 1, DP48	4		
DP39	BRANCH DIS 2, DP44	4		
DP40	BRANCH DIS 3, DP42	4		
	BRANCH DP17	4	Unconditional branch to DP17 (SUBR 3)	
DP37	BRANCH DIS 1, DP62	4		
DP56	BRANCH DIS 2, DP58	4		
DP57	BRANCH DIS 3, DP4	4		
	*Program section to process Frames 9-26 physically placed here reduces instruction bytes by 4.			
DP48	BRANCH DIS 2, DP54	4		
DP49	BRANCH DIS 3, DP50	4		
DP52	SET DIS 3	4		
DP53	SET IX 1, 4	3		
SPARES	FETCH SPARE, 2B, CC	4	Fetch 6 spares to round frame to 256 bytes	
	FETCH SPARE, 4B, CC, 5	4		
TAPE	BRANCH TAPE DUMP, YES	4	Check for Tape Dump Signal	
	ROUT	3	Route M-B, B-T but do not initialize.	
DP51	FETCH SYNC, H1, 4B, CC, 2I	4	Fetches SYNC, FRAME COUNTER, TIME and ID1	8 3 1
	BRANCH DP33	4	To SUBR 2	
YES	ROUT M-B, B-X	4	Route Buffer-to-transmitter to allow tape readout to ground.	
DP44	BRANCH DIS 3, DP4	4	To SUBR 1	
DP46	SET DIS 1	4		
DP47	SET IX_1, 8	3		
	BRANCH SPARES	4		
DP42	SET IX_1, 12	3		
DP43	SET DIS 2	4		
	BRANCH SPARES	4		
DP62	BRANCH DIS 2, DP63	4		
DP68	BRANCH DIS 3, DP69	4		
	BRANCH DP17	4	To SUBR 3	
DP58	BRANCH DIS 3, DP59	4		
	BRANCH DP17	4	To SUBR 3	

DP54	BRANCH DIS 3, DP55	4		
	BRANCH DP17	4	To SUBR 3	
DP55	FETCH SYNC, H1, 4B, CC, 2I	4	Fetch information to	8
	FETCH TIME, H3, 3B, CC	4	identify frame	3
	FETCH ID2, HS, 1B, CC	4		1
	BRANCH DP17	4	To SUBR 3	
DP50	SET DIS 2	4		
	BRANCH DP4	4	To SUBR 1	
DP63	BRANCH DIS 3, DP64	4		
DP66	RESET DIS 2	4		
DP67	SET IX 1, 12	3		
	BRANCH SPARES	4		
DP69	RESET DIS 1	4		
	BRANCH DP47	4		
DP59	RESET DIS 3	4		
	BRANCH DP53	4		
DP64	RESET DIS 3	4		
	BRANCH DP4	4	To SUBR 1	

SUBR 3

DP17	WAIT SS	2		
	RESET GATE A	4	Fetch "A" data	
	FETCH A1, 2B, CC, 2I	4		4
	SET GATE A	4		
	FETCH SPARES, 4B, CC, 2	4		8
DP18	WAIT SS	2		
	FETCH D1, 2B, CC, 3I	4	Fetch D and housekeeping	6
	FETCH H7, 2B, CC, 7	4	Data	14
DP19	BRANCH DIS 1, DP25	4		
DP20	BRANCH DIS 2, DP23	4		
DP21	SET DIS 3	4		
	BRANCH DP16	4		
DP25	BRANCH DIS 2, DP28	4		
DP26	SET DIS 3	4		
	BRANCH DP13	4		
DP23	RESET DIS 2	4		
	BRANCH DP11	4		
DP28	BRANCH DIS 3, DP31	4		
DP29	RESET DIS 2	4		
	BRANCH DP10	4		
DP31	SET DIS 4	4		
	BRANCH DP11	4		

End of section of program to process Frames 1-8.

3.10.6.3 Start of Section of Program Which Processes Frames 9-26

Detail Block No.	Instruction	(Bytes) Length	Comments	Data Bytes (Fetched)
DP72	SET IX 1, 17	3	Sets up for processing 17 frames (9-25)	
DP73	SET IX 2, 6	3	Sets initial stepping of filters after sectors of data are collected. Since 10 sectors have already been collected in the first 8 frames only 6 more are necessary before stepping filters (Every 16 sectors)	6
DP74	SET IX 3,5	3	Sets up a loop which causes 5 sectors of collected data to be formatted into 1 frame.	
DP75	BRANCH TAPE DUMP, DP76	4		
DP77	ROUT M-B, B-T	3	Route multiplexor-to-buffer-to-tape	
DP78	FETCH SYNC, H1, 4B, CC, 2I	4	Fetches the SYNC, FRAME	8
	FETCH TIME, H3, 3B, CC	4	COUNTER, TIME and ID2	3
	FETCH ID2, H5, 1B, CC	4	bytes to start a frame	1
DP79	WAIT SS/2	2	Idle until sub-sector/2 pulse is sensed	
	RESET GATE A	4		
	FETCH A1, 3B, CC, 5I	4	Fetch "A" data	15
	SET GATE A	4		
DP80	BRANCH DIS 2, DP81	4	DP81 is housekeeping fetch	
DP83	WAIT SS/2	2	Fetch "C" data	
	FETCH C1, 2B, CC, 7I	4		14
DP84	SET DIS 2	4		
DP85	WAIT SS/2	2	NOP - for proper timing between data fetches	
DP86	WAIT SS/2	2		
	FETCH D1, 3B, CC, 3I	4	Fetch "D" data	9
DP87	SET IX 4,7	3	Set IX reg for looping 7 times	
DP88	WAIT SS/2	2	NOP - timing delay	
	FETCH SPARE, 1B, CC	4	Fetch spares to round frames to 256 bytes	1
DP89	BRANCH IX4, DP88	4	Decrement IX reg 4 and branch back to DP88 until zero	
DP90	WAIT SS/2	2	Fetch "D" data	
	FETCH D1, 3B, CC, 3I	4		9
DP91	WAIT SS/2	2		
	WAIT SS/2	2	NOP	
	WAIT SS/2	2		
	WAIT SS/2	2		
	FETCH SPARE, 2B, CC	4	Fetch spares to round frame to 256 bytes	2
DP92	BRANCH DIS 1, DP93	4	Test for last frame processing	

DP94	BRANCH IX2, DP97	4	Check for time to step "A" filters
DP95	SET FILTERS DISCRETE	4	Step "A" data filters
DP96	SET IX 2, 16	3	Set IX reg. to count 16 sectors for stepping filters
DP97	BRANCH IX3, DP79	4	Test for frame complete (5 sectors)
DP98	BRANCH IX1, DP74	4	Test for completion of 17 frames
DP99	SET DIS 1	4	Indicator to program that last frame being processed.
DP100	SET IX 3, 1	3	Set IX reg for 1 sector in frame 26
	BRANCH DP75	4	Unconditional branch
DP76	ROUT M-B, B-X	3	Route multiplexor-to-buffer-to-transmitter
	BRANCH DP78	4	Unconditional branch
DP81	WAIT SS/2	2	Fetch HOUSEKEEPING
	FETCH H7, 2B, CC, 7	4	Data
DP82	RESET DIS 2	4	
	BRANCH DP85	4	
DP92	SET IX 5, 7	3	
	FETCH SPARE, 4B, CC, 6	4	
	BRANCH IX7, DP93	4	
	BRANCH DP2	4	

14

152 Instructions

555 bytes

6094 data bytes in 3 Rev.

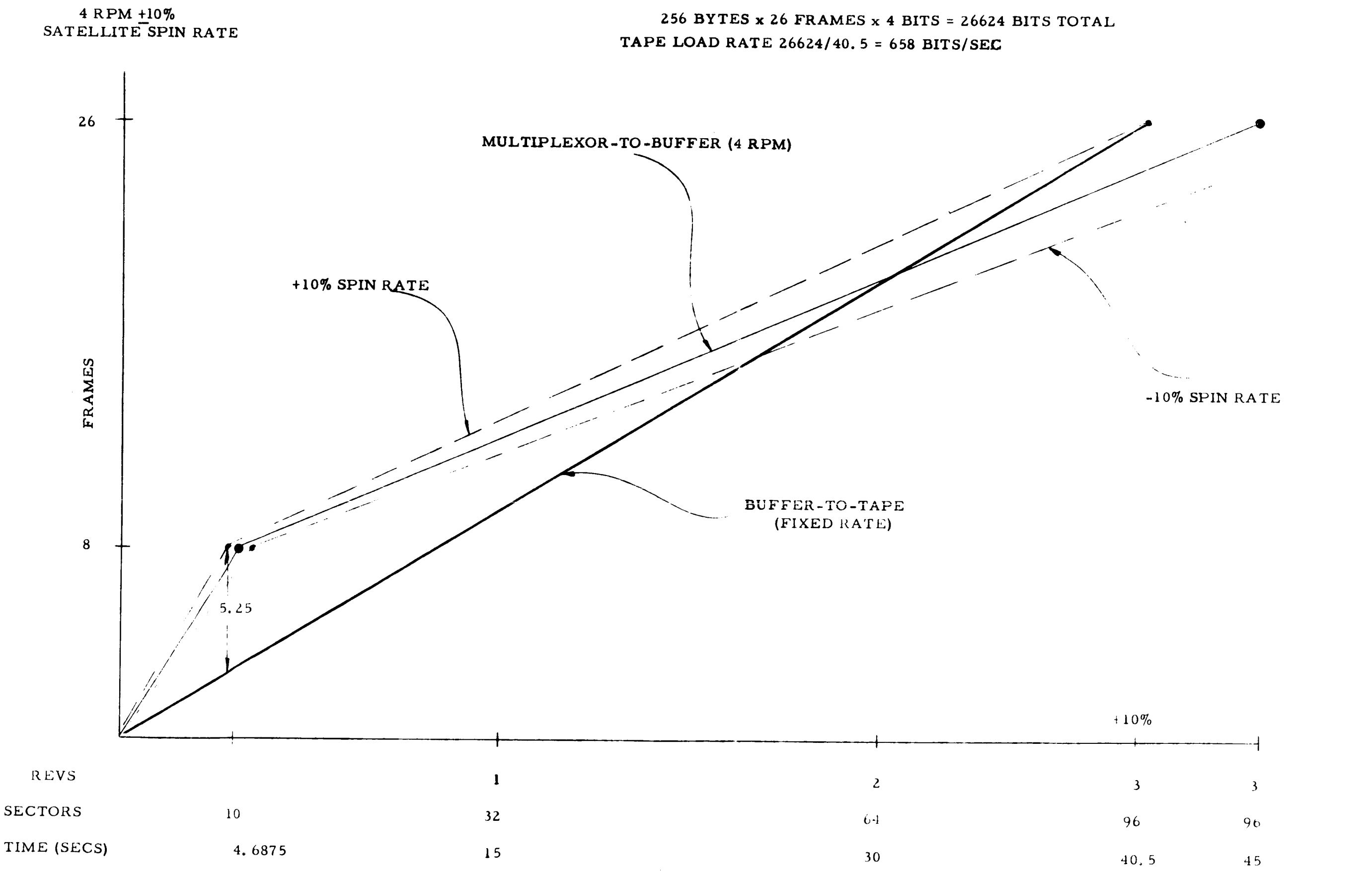
Tape Loading

The loading rate of the buffer memory and storage tape is shown in Figure 3.10-7. This loading is based on a 4 RPM ±10% spin rate for the satellite and a 3 revolution-26 frame program. (The program loads 26 frames of memory each complete pass and 3 revolutions of the satellite are required to complete each program pass.)

Since the tape speed is fixed and the data collection rate is dependent upon the satellite spin rate, the buffer memory between collecting the data and storing onto the tape becomes a major item in determining final memory size. The transfer of data from the buffer to the tape is automatic when the route (ROUT) instruction specifies Buffer-to-Tape. This means data is continually being transferred from the buffer to the tape at a fixed rate. However, data rate into the buffer is varying and this buffer loading may be faster or slower than the data retrieved by the tape storage. This implies that without proper controls, the data collection and subsequent storage onto tape would become jumbled and useless.

As shown by Figure 3.10-7, the buffer is loaded at a much faster rate initially (first 8 frames). The tape speed is selected such that at the completion of 3 revolutions of the satellite at the highest spin rate allowable (4 RPM + 10%) 26 frames of data will have been stored onto the tape. Using this scheme, it can be seen that if the data collection rate (multiplexor-to-buffer) is slower (Satellite spinning) less than 4 RPM + 10%), the 26 frames of data will not have been completed

Figure 3.10-7 Buffer I/O and Tape Load



as the tape loading requires and old data would be transferred to the tape. This is prevented from occurring by not allowing the OAR (Output Address Register) access to the same memory frame as the IAR (Input Address Register). If the tape attempts to fetch data from the same frame that is being loaded, the OAR is set to address the frame previously stored.

The optimum tape rate is 658 bits per second. Each frame of data contains 256 bytes with each byte consisting of 4 bits. The 26 frames contain 26624 bits (26 x 256 x 4). The buffer loading at the fastest roll rate is completed in 40.5 seconds (45 - 10%). Therefore, the tape speed should be 26624/40.5 or 658 bits/second. Assuming the tape storage capacity is $12(10^6)$ bits and a tape loading speed of 658 bits/sec., over 5 hours of storage time are available between tape readouts.

A slower tape loading rate would imply a longer storage time between tape dumps, however, the possibility exists that data could be lost, since at the higher satellite spin rate, data collection into the buffer storage would exceed transfer of the data to the tape. A faster tape rate naturally results in less time permitted between tape dumps and tape wastage. As can be seen from Figure 3.10-7, the buffer area must be six sectors or frames.

Memory Storage and Power

The core memory must be of sufficient size to hold the program instructions, the buffer memory, the hard-wired load program and index registers necessary (6 maximum).

The program described occupies 555 bytes and the buffer memory size requires 6 frames or 1536 bytes (6 x 256). The load program occupies sixteen bytes and four index registers are utilized which requires 8 bytes for a total of 2115 bytes. Therefore, a 4K memory is required. This provides adequate unused memory to permit program expansion or modification and additional buffer memory if necessary.

By utilizing various trade-off between program size and tape storage, it is felt a 2K memory could be utilized for the S^3A experiment. By utilizing a simple program approach, the instruction bytes can be reduced to permit a 2K memory usage. However, the tape storage utilization is reduced due to the number of spare bytes being transferred ($\approx$ 24%) and program expansion or modifications may be impossible since memory will be very nearly full

The study recommends a 4K memory for the experiment investigated.

In the course of the three roll program, 6656 total bytes of data are inputted to the buffer and 6656 total bytes of data are outputted from the buffer, in addition, 20,784 bytes of instruction fetching occurs. The total number of memory operations in three rolls is therefore 34,096. As three rolls take 45 $\pm$10% seconds, the worst average memory cycle time is 1,188μ seconds, which represents a duty cycle of only 0.84% (greater than 99% standby). Power consumption is very nearly that of standby power, or 1.73 watts (see section 3.3).

The total S^3 DPS system power for the S^3A system is as follows:

	Power
4K Memory	1.73
Central Control	1.18
I/O:	
Common	0.27
A/D Converter	0.50
Compressor	0.08
12 Analog 16 Digital Channels	0.07
Message Header	0.18
5 Accumulators	0.51
2 DOR's	0.08
	4.60 watts

While some components have been removed from the standard 4 delta pack design, the weight reduction is negligible; hence total weight of the S^3 DPS for the S^3A application would be 3.24 pounds.

4.0 TRADE DATA

A number of alternatives were considered in the evolution of the S^3 DPS design. These included variations in machine organization and instruction set, alternative logic design approaches, and alternative circuit and mechanical design technologies. The more significant of these alternatives are briefly described in this section.

4.1 Dual Program Machine

The single program machine described above, while it may contain several programs, can execute programs with either the DSC or the SCC clock stages but not with both concurrently. While it can satisfy all experiment sampling requirements, an alternative approach is a dual program machine wherein two programs, one for each clock system, can operate concurrently. Two programs are necessary for operating with the two clocks simultaneously because the two clocks are asynchronous - one is fixed, the other varies as the roll rate varies. As these "breath" one with respect to the other, so must the execution of the programs associated with them breath.

Accordingly, each program must have a separate program counter and buffer area, and each program must be executed on a clock interrupt basis. The interrupt system must store interrupts and assign one a higher priority, in the event the clocks occur simultaneously. It must also store interrupts until they can be honored. The delay in honoring an interrupt will depend on whether an interrupt process is in being and when it will finish. This is a function of how much processing is programmed for the various clock times. The S^3 DPS operates very fast with respect to individual clock sampling rates and data output rates, but when many things must be performed on the occurrence of a particular clock, a clock from

another source may be queued a noticeable amount. The same is true, of course, if higher speed clocks are used with less processing per clock.

The two programs - one for DSC and one for SCC sampling - must be separately stored in the DPS memory, and each must be assigned a separate buffer area. The events which must take place on the various clocks in each program must be preceded with a wait on clock instruction. When the machine is idle and waiting for either specified clock, it can accept either clock as an interrupt. A SCC(DSC) interrupt will cause the SCC (DSC) program counter to address memory so that the SCC (DSC) program picks up where it left off, and will inhibit the interrupt system. The SCC (DSC) program will then execute all instructions preceding the next Wait on clock. Upon fetching this instruction, the interrupt system is again enabled, and either clock will again be honored. If in the interim another clock had occurred it would be stored in the interrupt system and would be honored when the interrupt system is enabled.

Because two clocks may occur simultaneously, the interrupt system must assign one type a higher priority than the other. The priority assignment should not be fixed but should be user assignable to accommodate to the tighest jitter requirement. Since one would not program either the DSC or the SCC program to require more processing between its clocks than could be accommodated (this would be difficult to do anyway), one should not expect to find two clocks in interrupt storage at the moment it is enabled. At the moment of finishing a SCC (DSC) routine, one might expect to occasionally find a DSC (SCC) clock in storage; this might occur at regular intervals depending on the phasing of the clocks.

With this method of interrupt control, one program could lock out another for as long as about one millisecond. This would occur at those

times when the program, in addition to controlling sensors and acquiring burst data from a number of channels, also performs program housekeeping functions and acquires message header data before waiting for its next clock. (Lockout time could be reduced by about one-half by permitting interrupts at the end of instruction execution phases and permitting intermingled program operations; this was not examined in detail.)

Each program is assigned a buffer area of whole frames called sectors. Each is made as large as the difference between its peak data collection rate and the fixed output rate requires. Each buffer area is filled concurrently by the two programs, and the output address controls are set at the outset to correspond with the beginning of each buffer. Data output starts when either program has filled one frame (sector) of buffer. One of the buffers is normally read sequentially a frame at a time, being interrupted at frame boundaries to output one or more completed frames from the other buffer, before continuing on.

In implementation, the dual program machine differs from that described in seciton 3.0 only in the Central Control section where additional logic is necessary to permit concurrent operation of two programs, each with its own buffer area, on an asynchronous clock interrupt basis. A slight instruction set modification is also necessary.

The additional logic, the modified instruction set, and some of the characteristics of the dual program machine are discussed below.

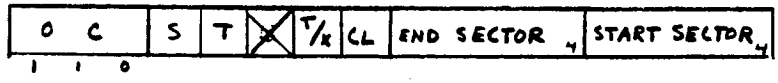
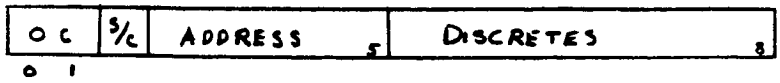
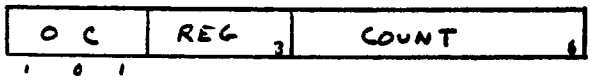
4.1.1 Instruction Set

To implement the dual program approach, some modifications to the instruction set are necessary. For example, the Command Chain bit in the Fetch instruction is used to hold up the execution of the Fetch until the next data sampling clock, but this capability can't be used in the dual program system because it could cause major program interference. The

Wait instruction would cause entirely unacceptable interference if one program put the machine in the idle state and the other program couldn't be fetched and executed. Thus this instruction must be implemented and executed in duplicate enabling one program to be in the wait state while the other is carrying on normal execution. This further complicates the I-fetch and Execute control phasing on a dual program system. An example of necessary additional instruction capability is in the initial program to set up buffer areas. Each program must have its own defined minimum and maximum buffer locations and a programmable instruction must be able to load buffer definition registers. This lengthens the Route instruction and complicates its implementation, particularly in the buffer output control area which it sets.

The dual program instruction set is described on the following pages.

INSTRUCTION SET

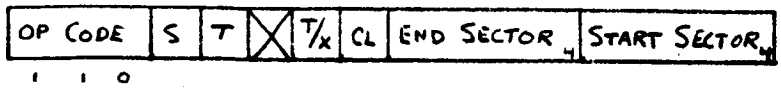
FETCH	F	
ROUTE	RT	
BRANCH	BR	
SET DISCRETES	SD	
SET INDEX	IX	
WAIT	W	

FETCH (F)

The Fetch instruction addresses the multiplexor to select data to be gated to the Route specified destination. This six-bit Channel Address is sent directly to the multiplexor to specify the data to be selected. The two-bit Word Length is also sent to the multiplexor; this specifies if the selected data should be 1, 2, 3, or 4 bytes in length.

The three Times bits indicate the number of times that the instruction execution is to be repeated. If the times bits are zero, the instruction execution will not be repeated. If the R/I bit is a zero (Repeat), the same Channel Address and Word Length bits are sent to the multiplexor. If the R/I bit is a one (Increment), the instruction will be repeated with the Channel Address incremented by one each time.

ROUTE (RT)



This instruction is used to initialize the Data Handling System (DHS) by setting up the data flow routes and defining the buffer areas available, thereby inhibiting data writing in the program area. The Tape bit (T) controls the tape recorder. When T = 1, the tape is to output data to the transmitter. Otherwise, T = 0 and the tape will receive whatever data is sent to it from the DPS. The T/X bit indicates if the data is to be routed from the buffer to the tape or transmitter. The Clock bit (CL) indicates the clock (DSC or SCC) to which the following sector bits refer. The four bit End Sector and Start Sector fields define the areas of the Buffer to be reserved for data collected by the DSC and SCC programs. These Buffer areas will not overlap. The S bit, set to a one, causes the initialization of the Input and Output Address Registers, if set to a zero, the sector fields are not sent to the Address Registers and the instruction just affects the routing.

BRANCH (BR)

The Branch instruction executes a simple branch if the condition specified is not satisfied. In branching, the ben bit Branch Address field is gated into the Program Address Register. In a conditional branch, if the condition is satisfied, the instruction stored after the branch instruction is fetched; i. e., it is a branch out of the loop on the specified condition.

Conditions 0 through 5 refer to index counters. When specified, the six-bit index word is read out of memory, placed in the Index Register, and decremented. The decremented index count is returned to its proper location in memory. If the count is not zero, the branch is executed. When the index count becomes zero, the condition is satisfied and the next stored instruction is fetched.

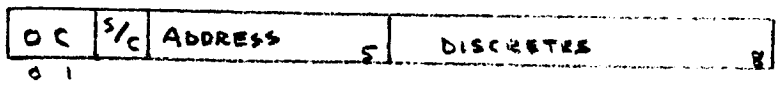
Condition 6 refers to the Mode Code Register (MCR). If this register contains all zeros, no branch is executed. If the MCR is non-zero, the MCR will be logically ANDed with bits 3, 4, 5, and 6 of the branch address. This address will be loaded into the Program Address Register. This gives a 16 way branch based on a ground command to the MCR.

Condition 7 is an unconditional branch to the specified address.

Conditions 8 and 9 are dependent upon the state of the first and second bits of the Discrete Input Register. This register may be set by inputs from ground control, on-board experimants, or on-board control and status monitors.

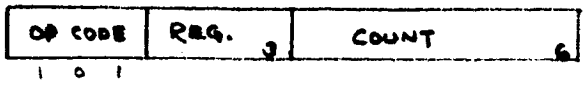
Conditions 10 through 15 are user connected to monitor whatever conditions may be of interest. These will be useful when connected to clocks, the Discrete Output Register, satellite mode indicators, experiment status monitors, and ground command registers.

SET DISCRETES (SD)



The Discrete instruction is used to turn on or off the individual stages of the Discrete Output Register. The S/C bit specified whether the selected register positions are to be turned on or off. The Address bits specify which eight bit register is to be acted upon and the eight Discretes bits indicate which bits of that register are to be set or cleared. There can be as many as five eight-bit discrete registers for controlling internal and external events. Discretes can be individually set and cleared, and as many as eight in one group, or register, can be set and cleared simultaneously. Additionally, the same bit position in several registers may be set or cleared simultaneously.

SET INDEX (IX)



This instruction merely initializes the contents of the index register specified by the Register bits. There are 6 index registers for each program, the DSC program and the SCC program. The index register specified, which is a location in memory, is loaded with the Count field. This index word will be fetched and decremented when a Branch instruction is conditional upon the index count. The status of the machine, whether in DSC or SCC mode, is used to double the total number of registers available.

WAIT (W)

The Wait instruction causes the program being executed to halt execution until the condition specified by the Condition bits is satisfied. This does not interfere with the execution of the other program. While the machine is in the wait state, data transfer out of the buffer to the tape will not be inhibited.

Condition 0 is System Clock 1 which may be user connected to whatever clocking frequency desired.

Conditions 1 and 2 are dependent upon the state of the third and fourth bits of the Discrete Input Register. This register is set by inputs from ground control, on-board experiments, or on-board control and status monitors.

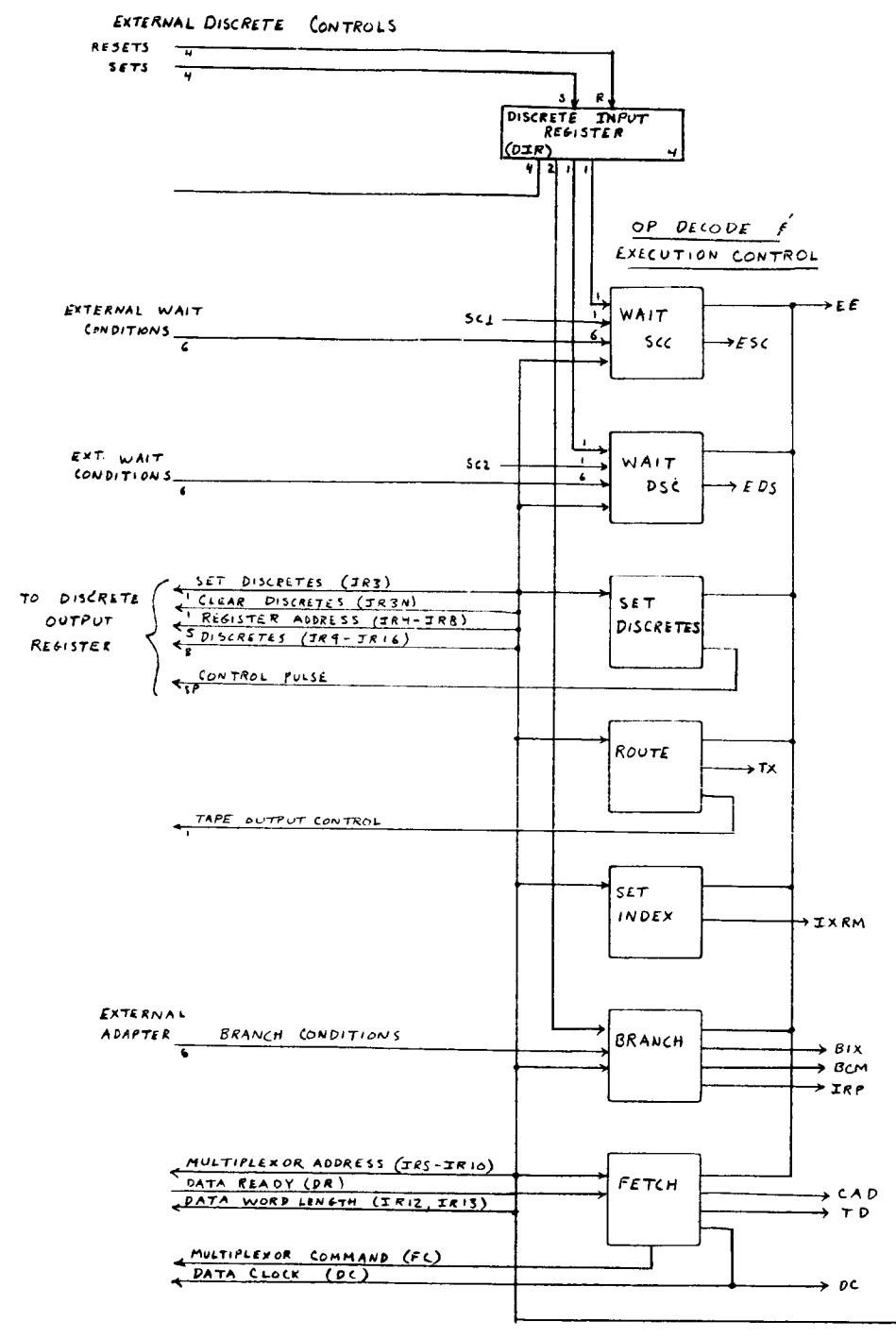
Conditions 3 through 7 are user connected to monitor whatever conditions may be of interest. This flexibility is similar to the user defined conditions available with the Branch instruction. Often these conditions will be connected to various timing signals.

4.1.2 Implementation

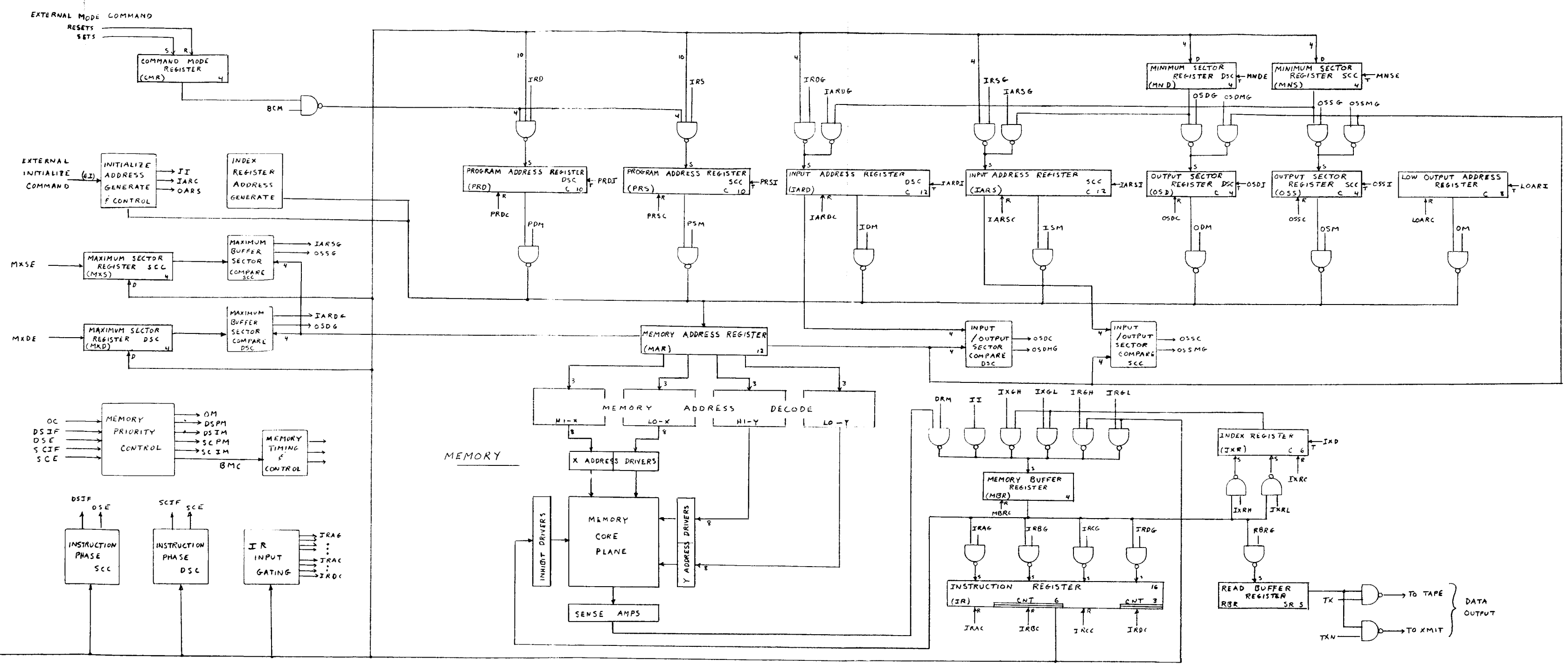
The dual program functional control and data flow is virtually identical to the single program approach with the exception of some areas being duplicated so that the program control can be separate for each timing system. One of the major areas of duplication in the dual program system is in program addressing and data input addressing where separate Program Address Registers and Input Address Registers are needed for each program. This is complicated by the necessity for defining minimum and maximum buffer areas for both the DSC and SCC buffer areas. Another duplicated area is in the clock generation system. This must have two independent systems for the dual program approach since both clocks will be used at once. Some instruction execution control is also duplicated. Other areas must be greatly enlarged and made more complicated to handle two programs. The memory priority control and output data addressing are examples of this. Figure 4.1-1 is a block diagram of the dual program system. This diagram points out the differences between the two systems when compared with Figure 3.2-2. Preliminary logic design of the dual program system indicates that this approach would require 167 more DTL flat packs and 582 milliwatts more power than the single program approach in the Central Control area. This is an increase of 50% in power and number of flat packs in the control. 167 flat packs are more than half of the components packaged in a delta pack so there would also be a significant increase in weight and volume.

As the total MIB area of the single program standard *design is only 31 flat packs short of filling the three delta packs, exclusive of memory, the dual program machine, less the accumulators, could be

*Near maximum

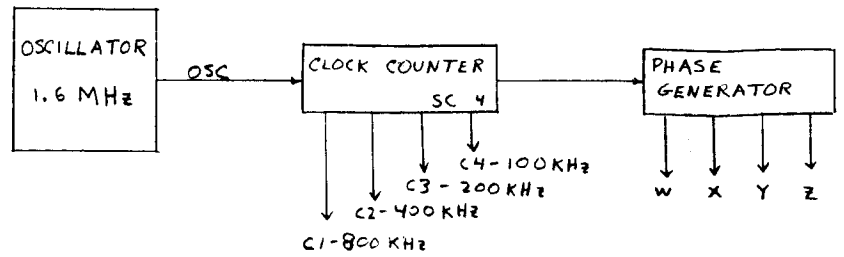


308-1



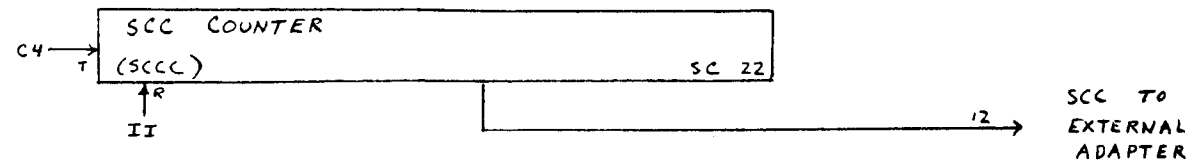
308-2

308-3

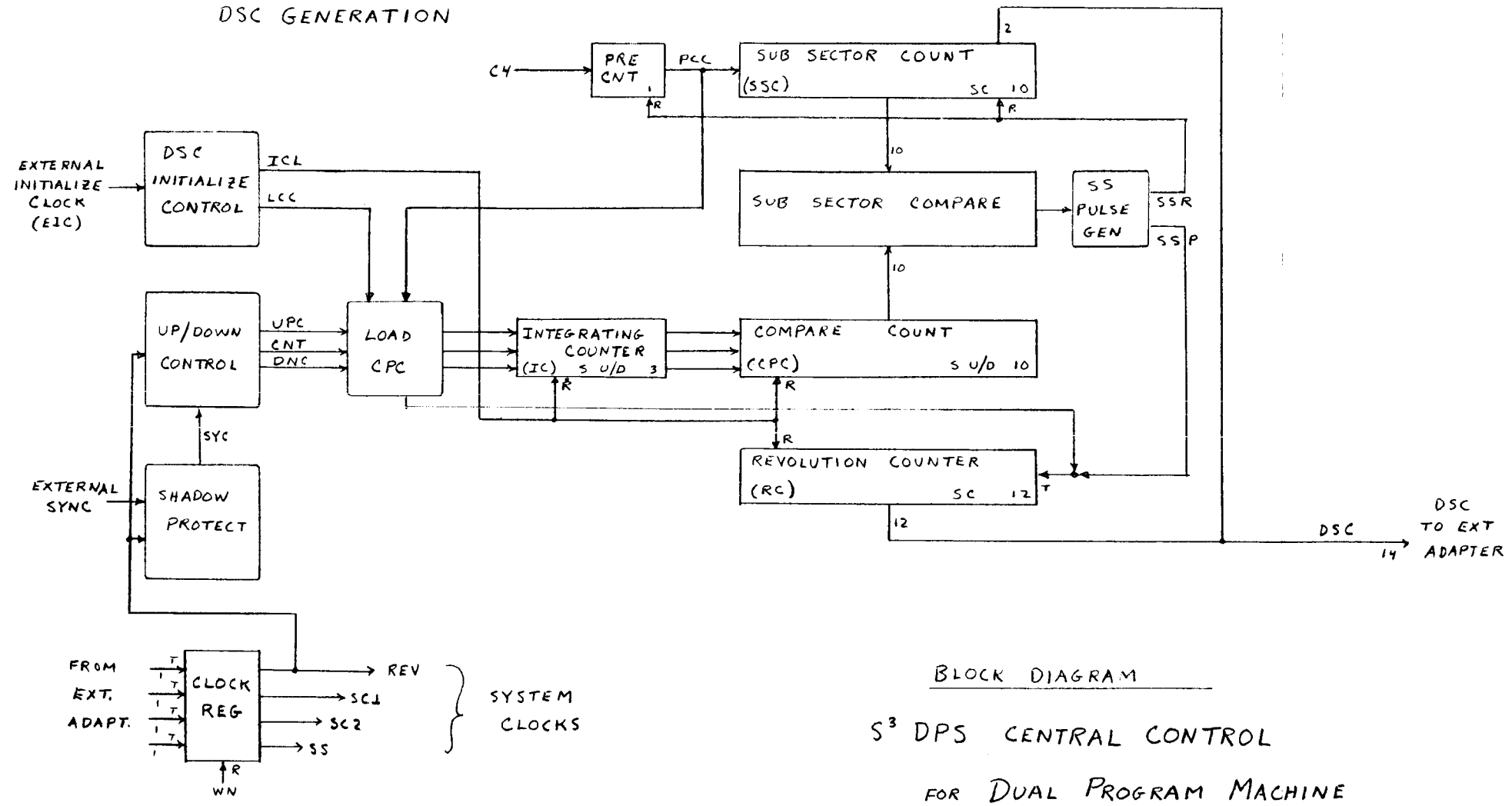


CONTROL TIMING

SCC GENERATION



DSC GENERATION



BLOCK DIAGRAM

S³ DPS CENTRAL CONTROL

FOR DUAL PROGRAM MACHINE

FIGURE 4.1 - 1

308-A-2

308 A -1

packaged in three delta packs. This would allow a 32 flat pack tolerance. If accumulators were needed, as in S^3A, an additional delta pack would be needed. Five accumulators would require almost one-half of a delta pack (section 3.4); a delta pack with a single MIB for the accumulators would add 0.42 pounds and one inch to the DPS.

4.2 Data Synchronized Tape Drive

Consideration was given to the concept of synchronizing the tape drive with the Data Synchronous Clock. The advantage of this concept is that the tape bit rate or bit density and transmission requirements would be reduced by about ten percent and the satellite spin rate would not need to be controlled to accommodate the tape. This approach to tape control was rejected by NASA mainly because their tape development program was felt to be too far along to implement any change.

The present system organization concept calls for a fixed output data bit rate roughly 10% higher than the nominal average input bit rate. This makes allowance for the variation in the DSC which would be a direct result of an error or change in the satellite spin rate. The spin rate must be controlled to not exceed 10% above nominal otherwise the tape would not keep up with the input data and frames of data would be lost. The tape unit uses a synchronous motor with ac power derived from a fixed frequency source.

Synchronizing the tape with the average nominal bit rate would require deriving the tape ac driving frequency from the DSC. This would eliminate the 10% safety factor in the tape rate, and proportionally reduce the tape storage and transmission capacity requirements. In addition, no reprogramming or satellite spin rate control would be necessary to prevent data loss if the spin rate were more than 10% off nominal. Since the tape must have an ac excitation frequency from some source, it could just as well be generated by the DSC.

One originally considered problem with using the DSC as the frequency source for the tape drive was the stability or rate of change of

the DSC. Referring to the section of this report on the DSC generation (3.2.6.3), it may be seen that any change in the DSC rate, in compensation for change in spin rate, will be less than 0.2% and this rate change would occur less frequently than once per minute. Changes in the actual satellite spin rate should be extremely slow.

Another consideration is that the data bit rate output to the transmitter must be constant regardless of changes in the data sampling rate or DSC. With both the tape speed and the DPS output data bit rate being directly proportional to the DSC, the bit density on the tape will remain constant. For example, if the satellite spin rate, and thus the DSC, slows down over some period of time by 1%, both the tape speed and the data rate to the tape speed will slow down by 1% since both would be directly controlled by the DSC. Thus the bit density which is data rate divided by tape speed remains constant. When the tape is read out for transmission to ground, it is speeded up and controlled by a fixed rate clock for generating the ac excitation frequency for the tape drive. Since the bit density is constant, this yields a constant output bit rate.

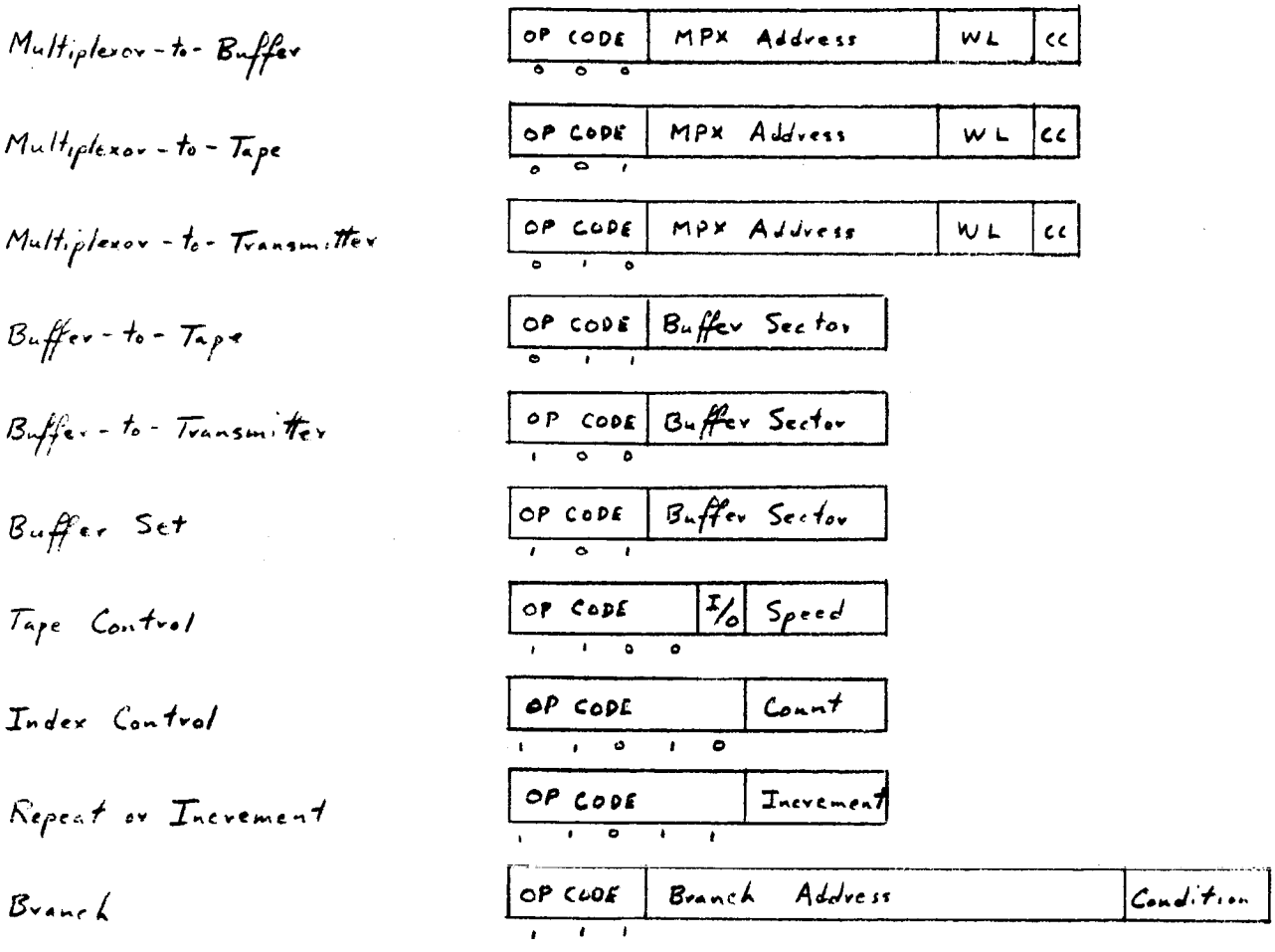
Actual spin rate is determinable from the program and the relative count field of the message header. In a three roll program, for example, the three roll interval is known to within 0.01 seconds simply from message header content. This equates to 0.05% for 4 RPM. Actual sampling rates within the frames collected over the three rolls is known to considerably better than 0.01 seconds of course.

4.3 Instruction Set

The evolution of the S^3 instruction set from the originally proposed set to its present form reflects a better understanding of what the S^3 experimenters really require and an optimization of the capabilities and implementation of the processor. The original instructions are listed in Table 4.3-I.

In the area of meeting experimenters requirements, instruction capability was added to provide communication and control with the experiment subsystems. A discrete output register, modular in blocks of eight bits to a maximum of forty bits, and a four bit discrete input register were added to provide this interface. The discrete outputs may be used to gate data inputs from sensors, set sensor modes, and notify other subsystems of existing conditions. Discrete outputs can be used to gate on and off the inputs to an accumulator, or many of them (up to 40) simultaneously, for example. They could also be used to step sensor modes or filters under program control. They can, in addition, be used by the processor as branch conditions so that one program can communicate to another for program linkage. The discrete input register permits experiment systems, vehicle systems, or the ground to communicate to the processor to cause it to branch to programs appropriate to the condition. If, for example, it is desired to acquire different data or data in a different way when a condition is sensed or at a particular time, setting an input discrete under these conditions will permit that change.

All of the original multiplex data fetching instructions were combined into a single fetch instruction with a repeating capability, with or without channel address incrementing, added to the fetch instruction. The repeat and increment capability had originally been implemented as



Original Proposed Instruction Set **Table 4.3.-1**

a separate instruction, but since the repeat or increment is used only with the fetch and the fetch quite often uses a repeat, an overall savings in program storage and execution control hardware was achieved. The fetch instruction length was made four bytes to incorporate the repeat and increment field of the separate instruction. A single Fetch instruction can now acquire up to four bytes each from up to eight consecutive channels or subchannels.

The specification of three routings out of the multiplexor in the original data fetching instructions is no longer necessary since all data must be first sent to the buffer. This necessity arises from the fact that various data sampling rates and word lengths are used and the tape recorder and transmitter operate at a continuous fixed bit rate asynchronous to the data sampling bit rate. It would be possible to provide a bypass of the buffer by direct multiplex to tape or transmitter routing, using the drop of the I/O Data Ready signal as the Fetch instruction execution clock and adding some logic for different data byte shifting control and paths. Data would then be acquired at a fixed byte rate; the byte rate (and the word lengths) would then determine sample rates rather than the other way around. However, not using the buffer presupposes that it is unusable, and if this were so, it would be very likely that the program would also be unusable. Therefore, this capability was eliminated.

The Route instruction replaces the Buffer-to-Tape, Buffer-to-Transmitter, and Buffer Set instructions. Since these three were all generally used together to set up routes and initialize the system, it was found more economical to combine them into a single instruction. The Route instruction specifies the memory assignment for buffer area for both inputting and outputting data address control. It can inherently

bypass a failed buffer area at the end of the buffer area, to bypass a failed area other than at the end it was given the capability to separately control input addressing without affecting the continuous fixed rate bit stream output.

The Wait instruction gives a new capability. It was implemented to save instructions and memory power while the machine was waiting on various conditions between data sampling. The most used wait conditions would be DSC and SCC clock stages to control the data acquisition rates or the times of execution of the appropriate program routines. The processor is a very high speed machine with respect to any single data acquisition or the data output rate, and must be so in order to perform simultaneous sampling or burst sampling and yet not affect the output bit rate; consequently, it is idle most of the time. The Wait on clock capability actually permits the machine to idle to save power and yet respond to output requests to maintain the constant output bit rate. A Fetch which is not command chained performs the same function as the Wait without requiring a programmed Wait, however, only one clock stage may be used for all unchained Fetches.

The Set Index is similar to that proposed but has a longer count field (six bits) and there are now six index registers, rather than one. These index registers are stored in memory and only one active hardware register is needed for indexing. An index register permits the same program segment to be repeated up to 64 times before branching out, and six index registers permit nesting of program segments six deep (loops within loops). The index registers permit a substantial reduction in program size and total memory size.

The Branch instruction is essentially the same as originally proposed except for additional branch condition hardware. The availability of discrete outputs to be tied back as branch conditions permits program linkage so that common subroutines can be linked to other program segments. Thus, the subroutines need only be stored once in one place in memory. Two discrete input register stages are also available for use as branch conditions. An added feature has been given to the branch instruction and that is a four-bit Mode Code. If the Mode Code is given as a branch condition, the four-bit code is used to modify the branch address, and the branch is unconditioned to the modified address. The four Mode Code bits are logically ANDed to four bits of the branch address. These four bits of the branch address can act as a mask to test the bits of the Mode Code. If the condition is not met, the four mask bits of the branch address becomes zero. If the condition is met, the branch will be to the address specified. If the mask bits are "ones", the branch will be to one of 16 addresses as specified by the Mode Code. The remaining address bits could be set to the address of a 64 byte block within which 16 4-byte unconditional branches to appropriate programs could be stored. The Mode Code would then provide a branch to one of 16 specified programs. This is a feature which can be partly or entirely deleted by flat pack removal.

Early in the design study, IBM solidified its proposed approach to use a single memory rather than two; one for the program and one for data. This was shown desirable on the grounds of power, weight, volume, flexibility, cost, and reliability. Independent of the consideration of a single or double memory, however, it is desirable to be able to bypass areas of memory which might have failed. Bypassing failed bits or words in the program area can be accomplished by unconditionally branching over the

failed area, and always was an inherent capability. To bypass failed bits or words in the buffer area requires a capability to skip over the affected sector or frame area. Since the tape unit operates continuously at a fixed rate, and operates asynchronously from the variable data acquisition rate, it cannot skip reading the affected frame area unless it is at the end of the buffer area. The new instruction set permits new frames of data to bypass the affected buffer area through programming. Only one stored instruction is necessary to bypass a buffer area, the new Route instruction. In this manner, no new data need be lost, and the tape (or transmitter) will still be presented a continuous fixed rate bit stream with synchronization codes at the same regular interval. Another alternative, of course, is to program blanks to the affected area. (Blanks could be any acquirable 4-bit quantity such as an ID code.)

4.4 Circuit Trade Data

4.4.1 DTL

Since very high speed is not essential in the performance of logic functions in the S^3 system, the prime tradeoff considerations in choosing a logic circuit family are weight, volume, power, and reliability.

From a power standpoint, only two circuit families offer a reasonable speed-power tradeoff compatible with the S^3 limitations. These are the Texas Instruments series 51 and the Fairchild low-power DTL, LPDTuL. The series 51 general logic circuits dissipate an average of 2 milliwatts per gate using a 3 volt supply. The LPDTuL general logic circuits dissipate an average of 1 mw per gate using a 5 volt supply. From the power standpoint, LPDTuL has the advantage over series 51. Also, the voltage magnitude required to achieve these low dissipations is in the favor of the LPDTuL family in this application because the higher supply voltage simplifies the problem of interfacing DTL with MOSFET technology as is required in this system.

A more significant advantage that the LPDTuL family offers is the logic flexibility that is obtained. This comes about partly because more efficient chips are available in LPDTuL than are available in series 51 and the fan-out of LPDTuL is twice that of series 51, but mainly because the series 51 RTL logic does not permit the multiple-logic-function per gate that is obtained in DTL from the collector dot capability. Each of these contributions to logic flexibility and efficiency reduces the flat pack requirements, and hence the weight and volume for the more efficient and flexible family decrease.

The series 51 circuit offers potential reliability problems because of the capacitor speed-up that is used on the resistor inputs. It would be expected that the circuits would be sensitive to impulse noise. The

straight-forward DC design of the LPDTuL circuit precludes this possibility and in addition the basic operation of the circuit to specification is not dependent on a difficult-to-control parameter such as an integrated capacitance. Because of the low fan-out of the series 51, the circuit reliability under radiation would also tend to be degraded because of the reduced margins in the design that is implied.

4.4.2 Functional MOSFET Devices

There are presently available from the General Instruments Company a number of functional MOSFET devices which could find application in the processor for the performance of logic functions. The circuit functions available include a dual three-input NOR, a four-bit up-down counter, and a number of shift registers of various length and input-output arrangements. These devices have the advantage that fewer devices are required to perform certain counting and shifting functions compared to the same implementation using DTL. However, the counters cannot be preset or parallel loaded.

To perform these functions requires considerably more devices than the name of a single device implies. The DTuL flip-flop is logically more capable than the MOSFET flip-flop, and requires less external logic than is obvious to implement useful functions. The major MOSFET logic gate is a dual 3-input gate while the DTuL family has a quadruple 2-input gate which is the most highly used gate arrangement in the S^3 design.

Implementation of the S^3 in existing functional MOSFET devices would require considerably more circuit flat packs with proportionate increase in weight and volume. Furthermore, the power required using the existing MOSFET devices would be greater than that required by LPDTuL. For example, the MEM 3012 is a 12-bit, serial in, serial out shift register which, though not a very useful function in the S^3 DPS, requires

162 mw average power. Twelve LPDTuL flip-flops dissipating a total of 48 mw can perform the same function, representing a power saving of better than three to one. The LPDTuL implementation offers the option of parallel inputs or outputs which the 3012 does not. Though there are parallel-in, parallel-out shift registers in the MOSFET family, the power required in every case is proportionally larger than that required by an LPDTuL implementation. It was concluded early in this study that the power advantage of the LPDTuL implementation plus its weight and volume advantage in implementing S^3 DPS functions was sufficient to outweigh the apparent weight and volume advantages of the functional MOSFET devices.

Implementation of the CCU and I/O with the existing MOSFET devices would result in higher power requirements by a factor of about 6 which would be totally unacceptable. The weight and volume requirements for a MOSFET implementation would be only slightly less than the DTuL implementation of the S^3 DPS. This is illustrated in the following table which was prepared prior to optimization of the designs and is for a maximum I/O system implemented with MEM 5015 multiplexing in each case.

DTuL

	MIBs	Power
CCU	2.83	1.13
I/O	3.17	3.84
	6.00*	4.97 watts

present MEMs

	MIBs	Power
CCU	3.08	10.38
I/O	2.39	18.0
	5.47*	28.38 watts

*a delta pack holds 2 MIBs

4.4.3 LSI Logic

At the present time, the only truly LSI logic that is available in the power range required by the S^3 system is the MEM 5015 16-bit random-access multiplexer (IBM 816 chip) and the MEM 5014 A/D converter (810 chip). The logic function to which the MEM 5015 may be applied is limited to matrix selection by four bits or less. The device consists essentially of four addressing flip-flops and 16 four-input AND gates.

For digital selection, this function in its entirety can be duplicated in LPDTuL with four flip-flops and eight dual four-input NAND gates requiring a total of 24 mw and 12 flatpacks. The 816 dissipates 50 mw and requires the mounting area of four flatpacks. However, the size and power comparison on these terms alone is incomplete. Because of the inability to perform logic functions on the input to the 816 flip-flops, logic circuits are required to develop the addressing prior to clocking the address into the multiplexer. Also level shifters would be required for interfacing the MEM 5015 to DTuL. In every case in this system, it has been found more efficient to address the digital gate or matrix element directly from the logic, thereby overcoming the size advantage of the 816.

For analog selection, or mixed analog and digital selection as is required in the I/O area, the MEM 5015 would seem ideal. For a 16 channel analog multiplexor, for example, the MEM 5015 with level shifters for DTuL interfaces would require a mounting area of 0.1 MIB's and 125 mw of power. However, using DTuL in conjunction with MEM 2009 devices and again including level shifters, the same function would require an area of 0.25 MIB's but only 40 mw. For digital multiplexing the DTuL implementation would require only 0.11 MIB's and 20 mw.

For a 32 analog, 32 digital channel multiplexor, including all level shifters and drivers the result would be:

	MIBs	Watts
MEM 5015	0.4	0.5
Hybrid	0.85	0.16

Thus the trade is between a third of a watt and potentially the weight of another deltapack. In the total system, however, space exists for the hybrid approach, hence it is used with no weight penalty and a savings of a third of a watt.

The MEM 5014 (810 chip) is the logic and switching portions of a 10-bit analog to digital converter. It and its associated level shifters occupy 0.07 MIB's and require 130 milliwatts. To replace the 810 in the A/D converter in this system would require 0.33 MIB's and 120 milliwatts using 4 MEM 2009s, 10 DTuL 9040's, 12 DTuL 9046's, 10 MD2402F transistor flatpacks, and 20 resistors. The trade data is illustrated below:

	MIBs	Watts
MEM 5014	0.07	0.13
Hybrid	0.33	0.12

The 0.26 MIB's is more significant than the 0.01 watts and hence the MEM 5014 is used in the A/D converter.

4.4.4 New LSI

Consideration was given to the possibility of implementing most of the functions in the S^3 system with new LSI devices tailored to the application and using the same technology as the 810 and 816 chips.

When tailored to the application significant power advantages can be had since the internal logic need only drive the internal loads encountered. A preliminary evaluation was made of the feasibility of using this technique. The design would have required the development of 15 new LSI devices of

the 810 and 816 technology. The devices were identified as to logic function, feasibility, and characteristics, and the development time was established as approximately three months. The cost to develop was averaged at $10,000 each. This included diffusion masks, jigs, dies, test fixtures, test procedures, and a small sample quantity of working devices. A preliminary estimate of components, power, and volume for four implementations was developed for the I/O section but not the CCU. The table below summarizes the results.

Complete I/O Section- Technology Comparison

	40 Pin Packages	14/16 Pin Packages	Misc	Mounting Area (Board Sides)	Power
Functional Mosfets	21	302	*	6	22 watts
Partial LSI	25	164	*	4	18 watts
Complete LSI	39	1	*	1 1/2	4.7 watts
LPDTuL, 810 & 816	26	307	*	4	3.8 watts

*Comparator, REF Voltage, Resistance Ladder, etc.

No further effort was expended in this area as it was considered beyond the scope of the intent of the study and not compatible with the desire to use available devices.

4.5 Word Length

Some consideration was given to the merits of using a three-bit byte rather than a four-bit byte. There are two major factors in this trade off, neither of which can be fully evaluated at this time:

1. Experiment accuracy requirements. Since most of the memory may be used as a buffer storage for experiment data, the memory byte length should conform as nearly as possible to the experiment requirements to keep from wasting memory and to conserve the telemetry capacity for fully useful information.

2. Power and hardware requirements. A shorter memory byte length requires more memory locations for an equivalent capacity. This implies more addressing hardware and more frequent memory accessing which requires more power.

A three bit byte organized memory is particularly advantageous if data words are 3, 6, 9, and 12 bits in length. The four-bit organization is preferable for words of 4, 8, 12, and 16 bits. For instructions, the four bit byte is more desirable because the instructions are in the 12 to 16 bit range due to the nature of the problem, and their length would be increased if more addressing were necessary as in a three bit byte memory. Due to the smaller bytes some additional logic would be required in the Central Control section. The instructions probably will account for only 25% of the S^3 storage requirements and virtually none of the telemetry requirements, except for infrequent reprogramming or program verification, but will account for about one-half the memory operations.

The experiment data length requirements cannot be fully defined for all S^3 missions although most types of data can be classified. Analog data accuracy requirements range from eight to ten bits. Eight bit data fits exactly into two 4-bit bytes but there is a one bit waste in a 3-bit organization. Ten bit conversion accuracy will imply wasting 2 bits

for either the 3 or 4 bit byte organization. For analog data, therefore, the 4 bit organization is preferable.

A number of experiment sensors will generate randomly spaced pulses for accumulation over fixed periods of time. The accumulated samples which may be up to 20 bits in length are logarithmically compressed to conserve both memory and telemetry bandwidth while preserving accuracy. The required compression is to 8 or 9 bits. The 9-bit compressed data would represent the compression of a 20-bit word with a maximum error of 3%. The 8 bit conversion is to be used to compress 12-bit words with 3% maximum error (or 20 bit words which have little difference from sample to sample and whose range is known). The eight bit compression is partly to conserve tape storage. For a 4-bit byte organization 8 bits is ideal but 9 bits would represent a waste of 3 bits per sample. (Actually, for 8 flatpacks more, the compressor could supply 12-bit data for a maximum error of 0.39% for 20 bits). For the three-bit byte organization, 9 bit compression is ideal, but 8-bit compression would waste one bit per sample. As 9 bits per sample is expected to be most prevalent, the three-bit organization is better for compressing 20 bits with 3% accuracy. If compression to better than 3% were required, the four bit organization would be better.

In S^3A as presently defined, the requirement is to collect 15,470 bits in three revolutions, exclusive of housekeeping and message headers. A 3-bit byte organization would have to collect 16,930 bits and a 4-bit byte organization would have to collect 18,190 bits to strictly comply with sampling requirements. All else equal, the four bit organization would transmit about 7.5% more bits than the three bit organization for S^3A as presently specified. In application this difference could be either increased or decreased. Memory buffer operations would be 9,095 vs. 11,300 for

4-bit and 3-bit organization, respectively. In practice, due to fixed frames and frame lengths, these figures would be increased and not necessarily by the same percentages. For the 4 bit organization the memory required for S^3A is 2,115 words for buffer and program; whereas for a 3 bit organization the number of words required would be 2,820 words or more (a 33% increase). Similarly, the memory duty cycle would be increased about 33% for the program portion in a 3 bit organization. Because a 3 bit organization would more often require the larger size memory and would require more logic the 4 bit word would be preferable.

4.6 Memory

The memory study consisted of investigations of ferrite cores and magnetic film elements, each type of element being considered for DRO and NDRO properties, power, size, and reliability.

The study was based on information obtained from IBM development experience on the Orbiting Astronomical Laboratory, Gemini and Saturn V contracts, and on reports from IBM advanced development laboratories. Industrywide memory announcements available in the literature were also examined.

Alternative memory elements and configurations considered is discussed below.

The decision to use a toroidal ferrite core memory with the configuration before described is the result of a number of factors enumerated below.

1. Ferrite cores in general have proven their extremely high reliability through years of use. The particular core to be used is one of IBM's advanced lithium-nickel alloy cores with a small temperature variation in switching threshold, requiring minimal temperature compensation of drive currents.
2. In general the coincident-current method of addressing requires a smaller amount of electronics than the word-organized method. The coincident-current scheme is especially well adapted to cores in this application, because extremely high speeds are not required.

3. Because of no stringent speed requirements, extremely fast rise times on drive currents are not required. This allows simple, straightforward electronic circuits comprised of DTL integrated circuits, hybrid multiple-diode and transistor flatpacks, and discrete components. Practically every component has already been qualified for aerospace environments.

4. The core plane frame is an existing design developed on NASA contracts for the Saturn V and the Apollo Backup Computer programs. This will allow a significant saving in overall development cost.

4.6.1 Ferrite Cores

Two types of toroidal cores, both manufactured by IBM, were considered. One of these was a 30-mil (O.D.) core used in the Saturn V computer memory, the other a 21-mil core used in the 4 Pi computer memory.

Over the expected memory core temperature range of -10 to +35°C, the 21 mil core requires less switching energy and has a much flatter switching threshold variation with temperature, therefore requiring a simple temperature compensation circuit. The 21 mil core was the final choice for the S^3 memory.

Dual-aperture cores such as the IBM "MARS" (Multi-Aperture Reluctance Switch) used in the memory for the IBM on-board computer for the Gemini program can provide NDRO operation. However, the MARS device is physically large and requires higher switching energy than the toroidal cores mentioned above. Other dual aperture cores which were

surveyed required drive current rise times nearly an order of magnitude faster than those proposed for the S^3 memory, therefore complicating driver circuit design. Overall, the NDRO type cores appeared undesirable because either the complex drive circuitry or the higher switching energy requirements would most probably add to the weight, volume, and power of the memory.

4.6.2 Magnetic Film Memories

Magnetic film elements may be classified into (1) cylindrical films and (2) flat films.

Cylindrical film elements are usually formed by continuously plating a thin layer of nickel-iron film on a copper wire in the presence of a circumferential magnetic field which magnetically orients the film. An "easy axis", which is the preferred axis of magnetization, is formed around the circumference of the wire. The plated wire is cut into required lengths which are used as bit/sense lines. Word lines in the form of thin straps or woven wires are located orthogonal to the plated wires.

NDRO operation is achieved by passing a current pulse through the word line. This induces a transverse magnetic field which rotates the magnetic vectors in the film element less than 90 degrees away from the easy axis, inducing a voltage in the bit/sense line which is amplified by sense smplifiers. The polarity of the voltage depends on the original direction of the magnetic vector. When the word current is removed, the magnetic vectors return to their original positions. Writing is accomplished by a sequence of word and bit pulses which rotates the magnetic vector more than 90 degrees through the hard axis to the desired polarity in the easy axis direction.

IBM advanced development laboratories conducting tests on single layer continuous plated wire films have observed various problems which are enumerated below.

1. Bit Spreading

   Repeated writing of the same polarity into one location enlarges the bit. The magnetic field spreads over to adjacent bits on the plated wire and reduces their amplitudes. This phenomenon can be reduced by accurately controlled bipolar write currents or other means.

2. Corrosion

   A humid environment can cause corrosion of the plating with consequent loss of bits if no protective coating is placed on the wires.

3. Film Aging

   With time, temperature, and use, the film has been observed to lose the distinct orientation of easy and hard axes and tend to become isotropic. One result of this is to gradually reduce the amplitudes of the output voltages until the memory no longer operates. Another result is that as the film ages, the optimum word current amplitude and pulse width change. Therefore a point may be reached where the word currents can cause DRO operation with consequent loss of memory words. High temperature annealing appears to alleviate the aging problem considerably. IBM is investigating several approaches to solving these problems while maintaining ease of manufacture, low cost, and reliability.

Among the approaches being investigated are coupled film elements. The memory elements are deposited on the cylindrical substrate in discrete rings to eliminate bit spreading. Annealing plus various film compositions are being investigated to reduce aging to a minimum. A conformal coating over the completed elements eliminates corrosion.

These approaches, IBM believes, will yield a more reliable NDRO device than a single-film element.

Flat-film memories operate in much the same manner as the cylindrical film devices. Discrete spots or continuous layers of film are sandwiched between orthogonal word and bit/sense lines.

Several manufacturer, including IBM, have announced and/or delivered flat-film memories of various sizes and speeds.

Flat-films have been used for aerospace memories and have demonstrated reliable operation. However, they appear much better suited for large, mass-storage ground use because the cost per bit drops drastically as the memory size increases.

4.6.3 Conclusions

In summary, the memory survey has led to the conclusion that the best solution to the size, weight, power, reliability and cost requirements for the S^3 memory, in view of the delivery schedule for the earlier missions, is a ferrite core memory.

IBM expects that in the near future it will be able to offer a cylindrical film aerospace memory with a reliability as high as that of the core memory, and with decreased power requirements, which could possibly be phased into later S^3 missions.

4.7 Alternate Mechanical Approaches

The Deutsch RM connector is not ideally suited for the electrical interface connector for the Data Processing System delta pack subassembly designs described herein. This opinion is based upon the following considerations:

1. The Deutsch RM connector is designed to accept a discrete wiring harness at both the connector plug and receptacle ends. As such, a jumper wiring harness must be used between the MIB's and the male half of the connector mounted on the Data Processing System delta packs.
2. The overall delta pack structure height of the Central Control and I/O subassemblies is increased because it is impractical to mount the Deutsch RM connector on the center line of the structure web since the structural integrity of the delta pack would be impaired and the component mounting area reduced.
3. The overall system reliability is reduced due to the increased number of soldered connections necessary to provide jumper wire connections between the MIB's and the Deutsch RM I/O connectors.
4. Subassembly fabrication is made more difficult due to the short length of the jumper wire leads and the restricted area in which to dress and solder them in place.
5. Strain relief must be provided for the jumper leads at the rear surface of each I/O connector by means of conformal coating or encapsulation to prevent damage during handling or under vibration conditions.

In the interest of utilizing a standard type I/O connector throughout the S^3 satellite, GSFC established as a study design ground rule that the Deutsch RM connector would be utilized for all delta pack subassemblies. However, at a meeting at GSFC on 23 February 1967, IBM personnel described, as part of a mechanical design presentation, the development history and application of the 98 pin Burndy ML 98 P5 connector to provide electrical I/O interface to multilayer interconnection board subassemblies. As a result of this presentation, IBM was requested to prepare a layout of a delta pack subassembly utilizing the Burndy connector and to discuss its application in the final report.

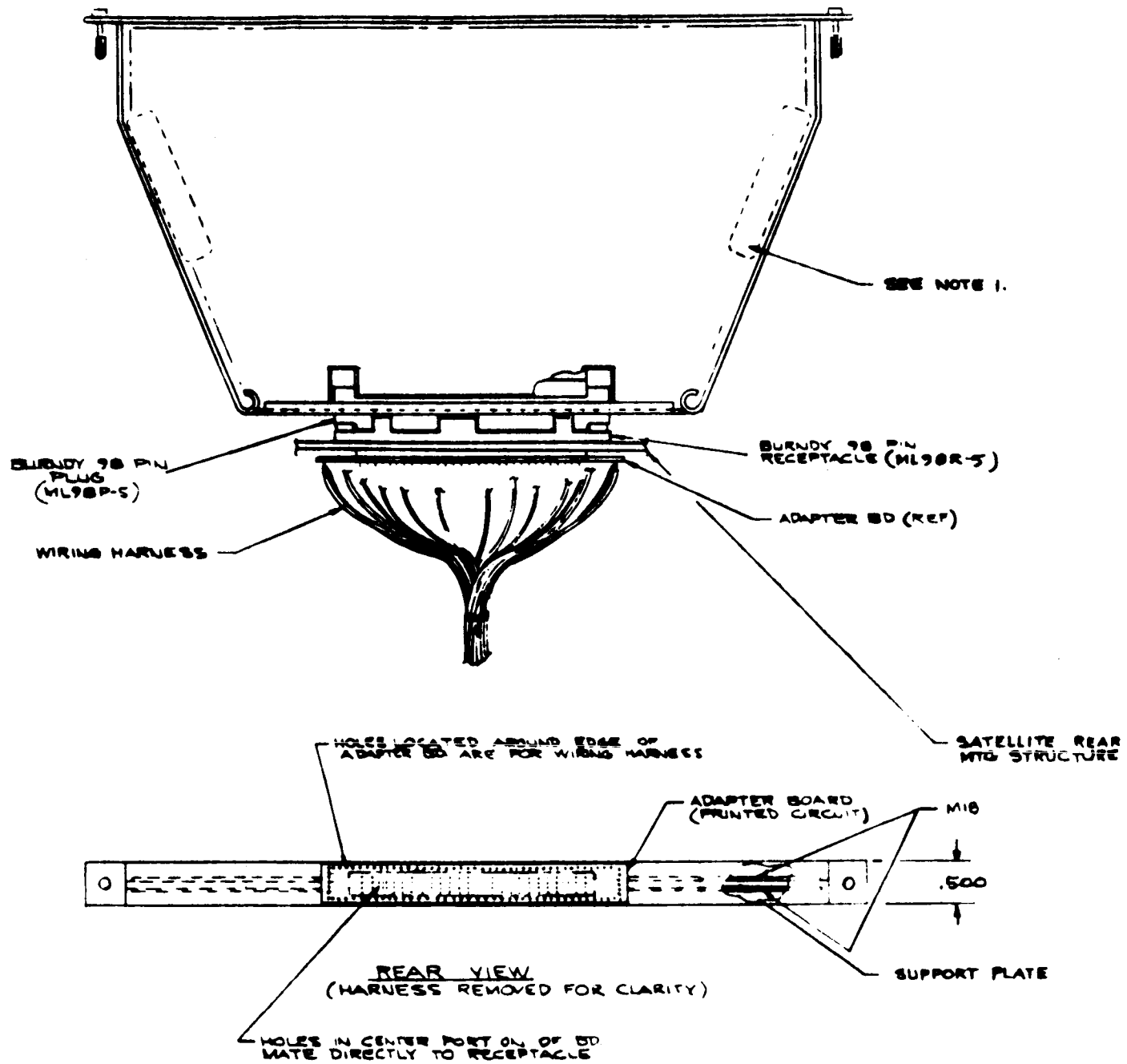
The use of the Burndy ML 98 P5 connector in a delta pack structure is described in Figure 4.7-1. An alternate arrangement using a Burndy Type UPC connector is described by Figure 4.7-2. The advantages of these design approaches are enumerated below.

1. The overall height of the Central Control and I/O delta pack assemblies can be reduced by 50% through the use of the Burndy ML 98 P5 connector and by 25% using the Burndy UPC connector.
2. The weight of each of the above delta pack frames will be reduced by 20% or by 16% using the ML 98 P5 or UPC connectors, respectively.
3. The system reliability will increase due to the elimination of the I/O connector to MIB jumper leads.
4. MIB component mounting area is increased by 10%.
5. Standard mating receptacles are available for each connector, plus a variety of sizes for the Burndy UPC connector. Typical

connector/receptacle mounting arrangements are shown in Figures 4.7-1 and 4.7-2. No additional mounting area is required for these mating receptacles compared to the Deutsch receptacle.

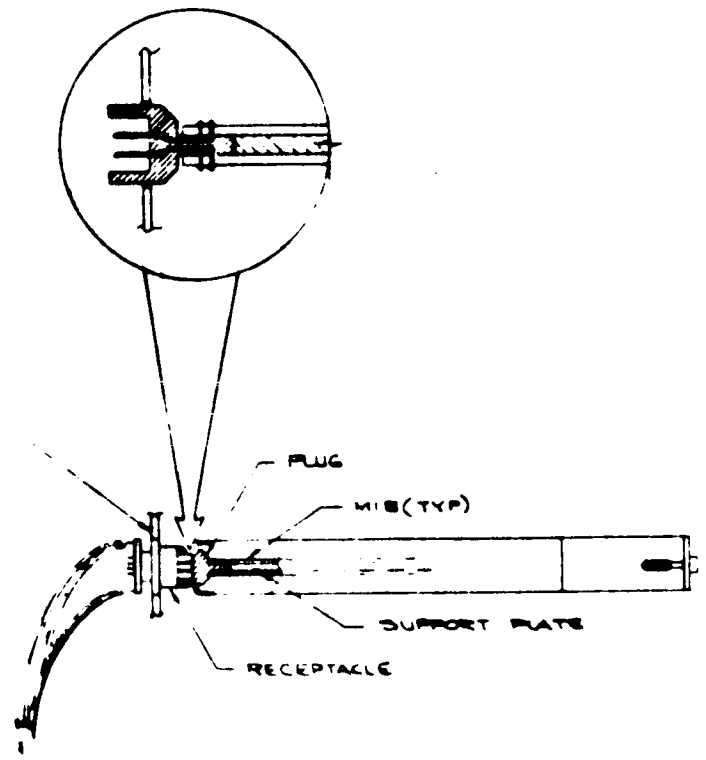
6. The Burndy ML 98 P5 connector and ML 98 R5 receptacles were developed for the electronic page application during the Saturn V program and provide I/O electrical interface to two multilayer interconnection boards simultaneously. These connectors are currently being used as the electronic page connectors in IBM's 4 Pi computer family. Also this connector receptacle combination is used as the page to I/O harness electrical interface in IBM's Model TC computer. The Burndy UPC connector/receptacle combination is utilized in the S^3 DPS Memory Unit design to interconnect the memory array to the multilayer interconnection board, and it is used for a similar application in the 4 Pi computer families.
7. Both the Burndy ML 98 P5 and the UPC connectors have been qualified for space applications.

The above discussion is offered as an alternate design approach to the problem of interconnecting the delta pack I/O connectors to MIB's as used in Data Processing System design, and is not intended to detract from the suitability of the Deutsch RM connector selected by GSFC for other delta pack design configurations. It is believed, however, that the Burndy printed circuit board connectors described above offer significant weight, volume, and reliability advantages over the Deutsch connector for use in the Data Processing System subassembly designs and it is recommended that GSFC further consider the use of these or other similar connectors for application to the S^3 program.

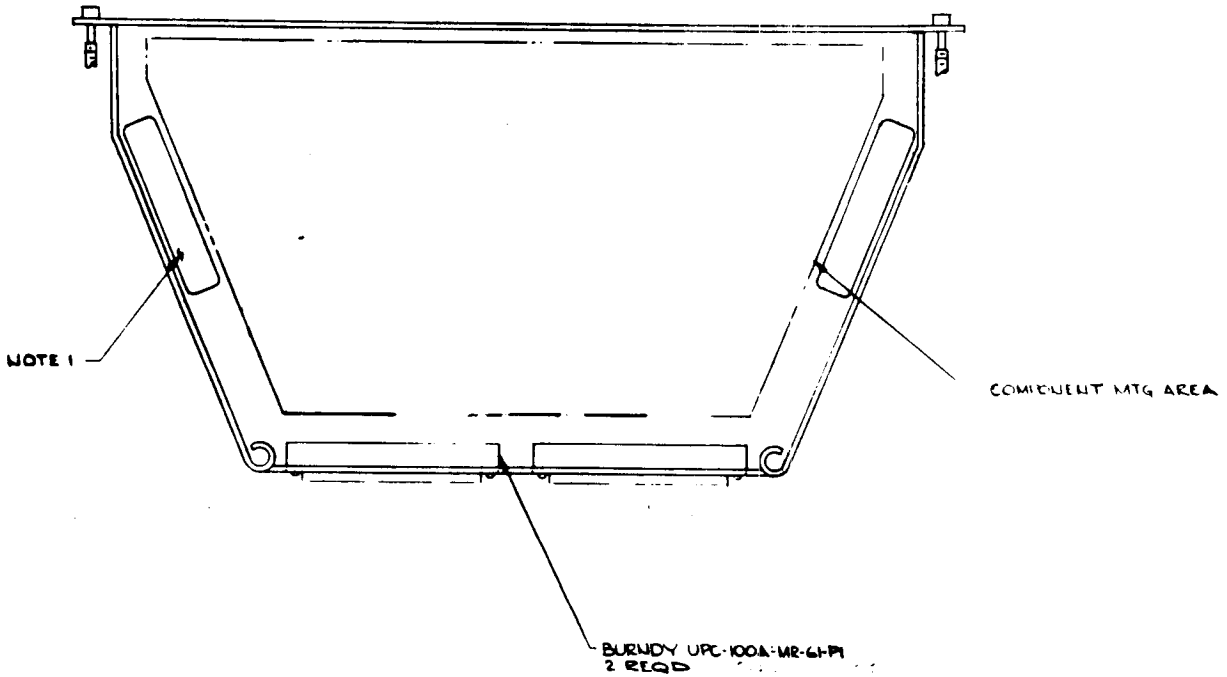


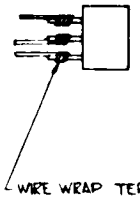
NOTES:
1. TEST POINTS AND OR THRU PIN AREA
2. DELTA PACK HEIGHT COMPARISON -
   (A) DEUTECH CONNECTOR METHOD HEIGHT - 50% LESS
   (B) BURNDY UPC CONNECTOR METHOD HEIGHT - 35% LESS

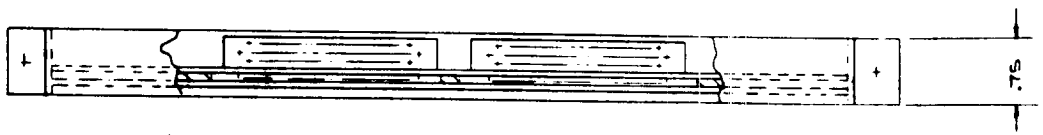
FIGURE 4.7-1

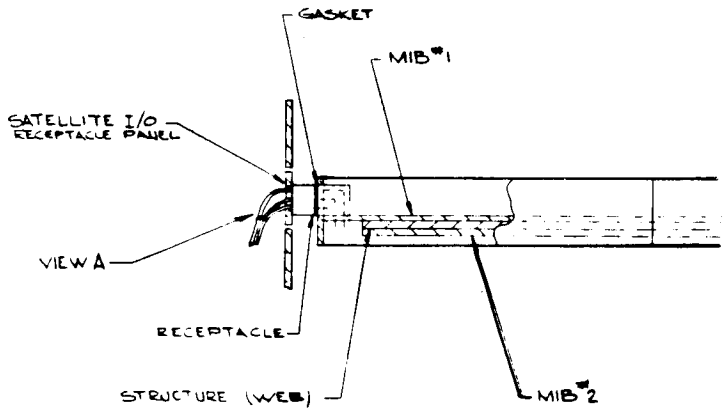


G³ DELTA PACS WITH BURNDY 98 PIN CONNECTOR AND ADAPTER BOARD









NOTE
1. TEST POINTS AND/OR THRU PN AREA
2. HEIGHT / WEIGHT REDUCTION COMPARED TO DEUTSCH CONNECTOR METHOD
   HEIGHT - 25%
   WEIGHT - 11% WT/STRUCTURE

FIGURE 4.7-2

S³ DELTA PACK WITH 2 BURNDY RIGHT ANGLE I/O CONNECTORS PC-DCA-MR-61-P1

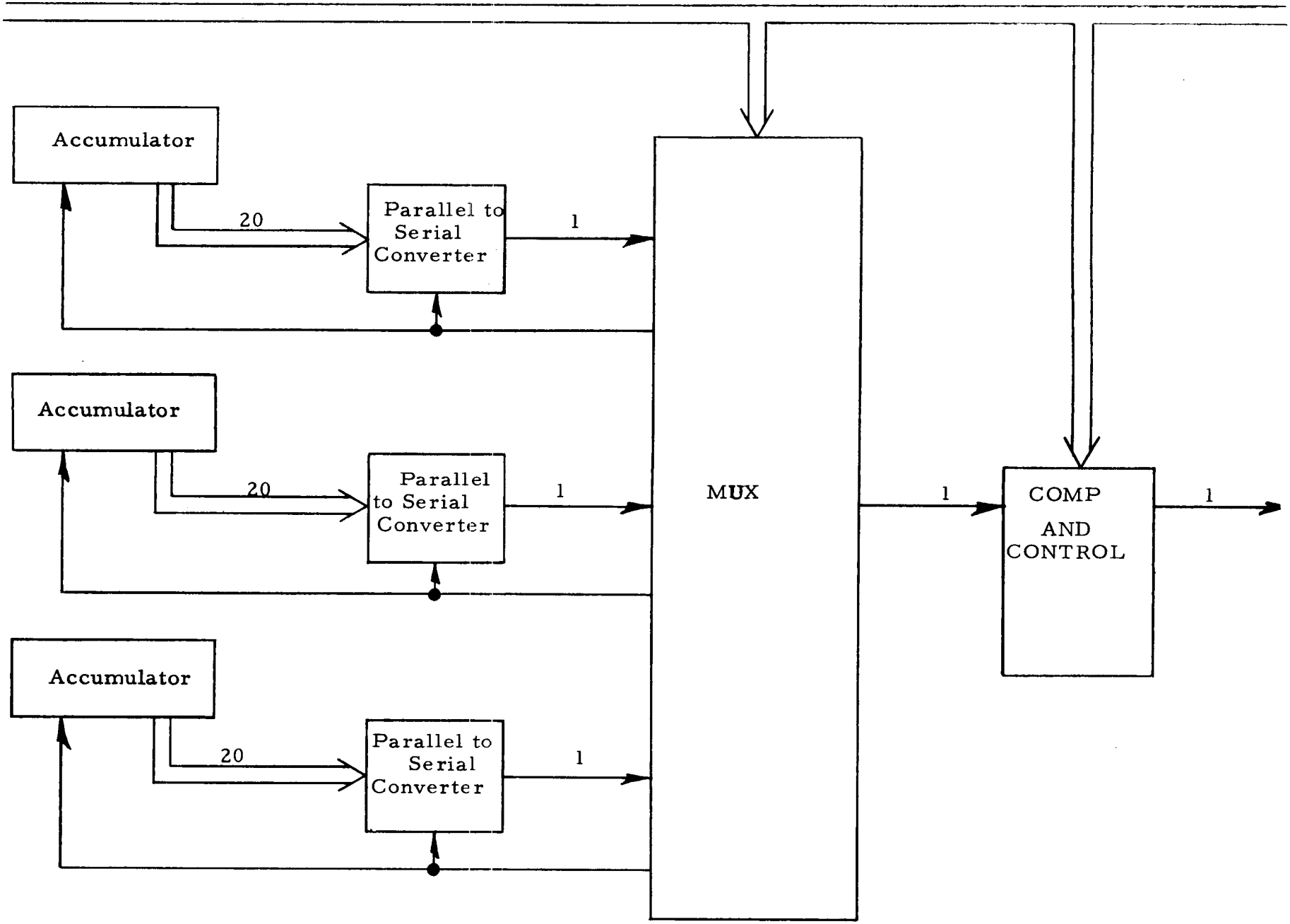
MOI-0070 SCALE : 1 / 1

4.8 Accumulators and Compressors

4.8.1 Approaches

The multiplexing of analog voltages and single ended digital data sources presents no real problems; the accumulation of up to 20 bits of data, the selection of an accumulator, and the logarithmic compression of 20 binary bits to nine bits does. Figures 4.8-1 through 4.8-5 show ways by which the process can be implemented. One aspect of flexible design is uniformity. If all digital data sources could be made to look alike, then it would not matter how many of what kind were connected to the system. Therefore, the approaches illustrated in Figures 4.8-4 and 4.8-5 were eliminated.

When all LSI was still being considered, Figure 4.8-3 represented the minimum volume approach. This approach uses static compression which is readily implemented in 40 pin LSI packages. Each accumulator/compressor channel would require but three packages: a 20 stage parallel-out counter, 20-input/9-output compressor, and a 9 stage parallel to serial converter. Dynamic compression would be more complex to implement in LSI technology. The static compressor is illustrated in Figure 4.8-6. Here all signals are generated simultaneous using only Nand gates (a static process). The bit positions of the accumulator gated into the 5-bit binary value is a function of the exponent and is shown in Table 4.8-1. The dynamic compressor involves counting and shifting. Details of the dynamic compression process are discussed in section 3.0.



Multiplexer Configuration With Input Parallel/Serial Conversion, Output Compression
Figure 4.8-1

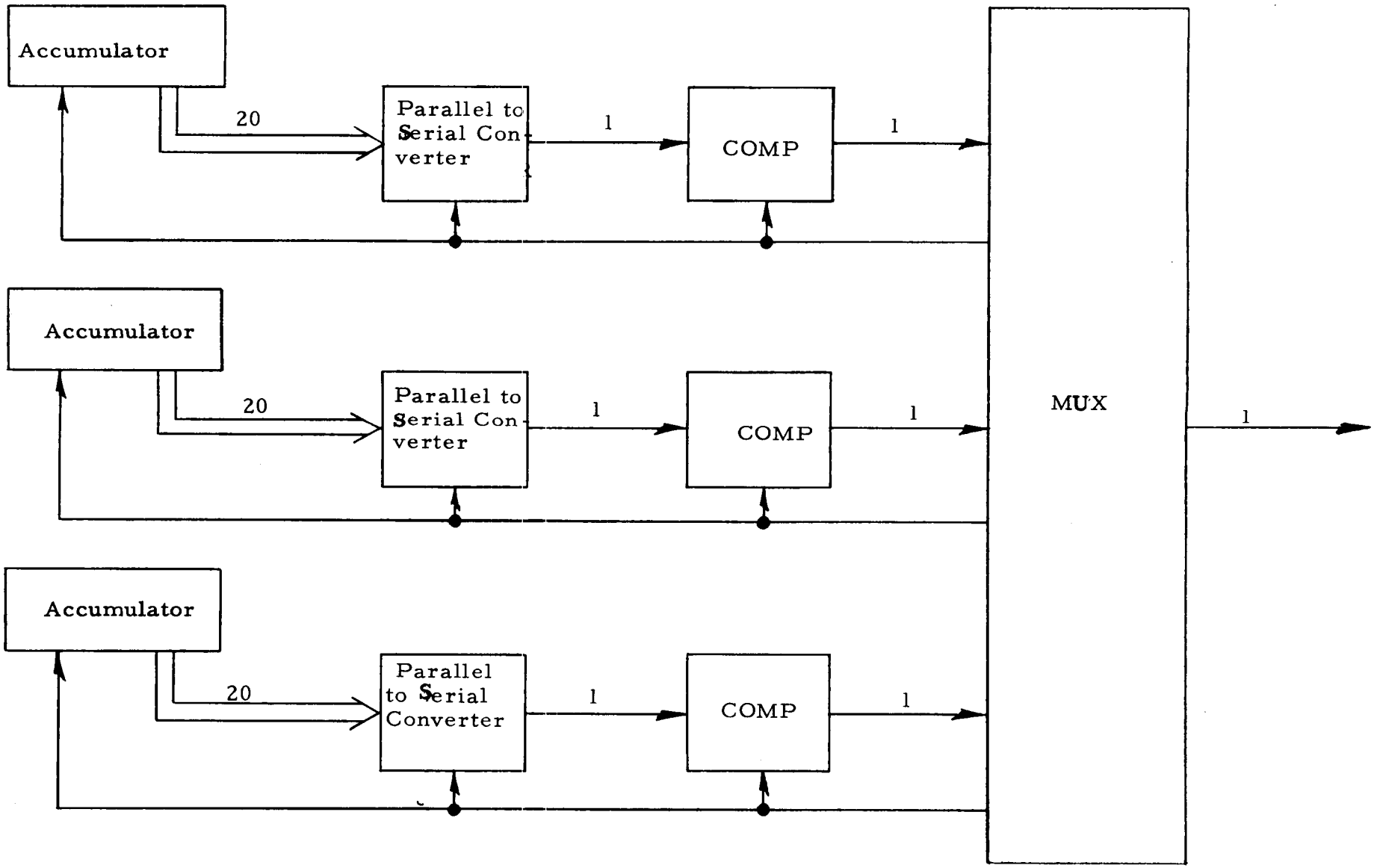


Figure 4.8-2

Multiplexer Configuration With Input Parallel/Serial Conversion and Compression

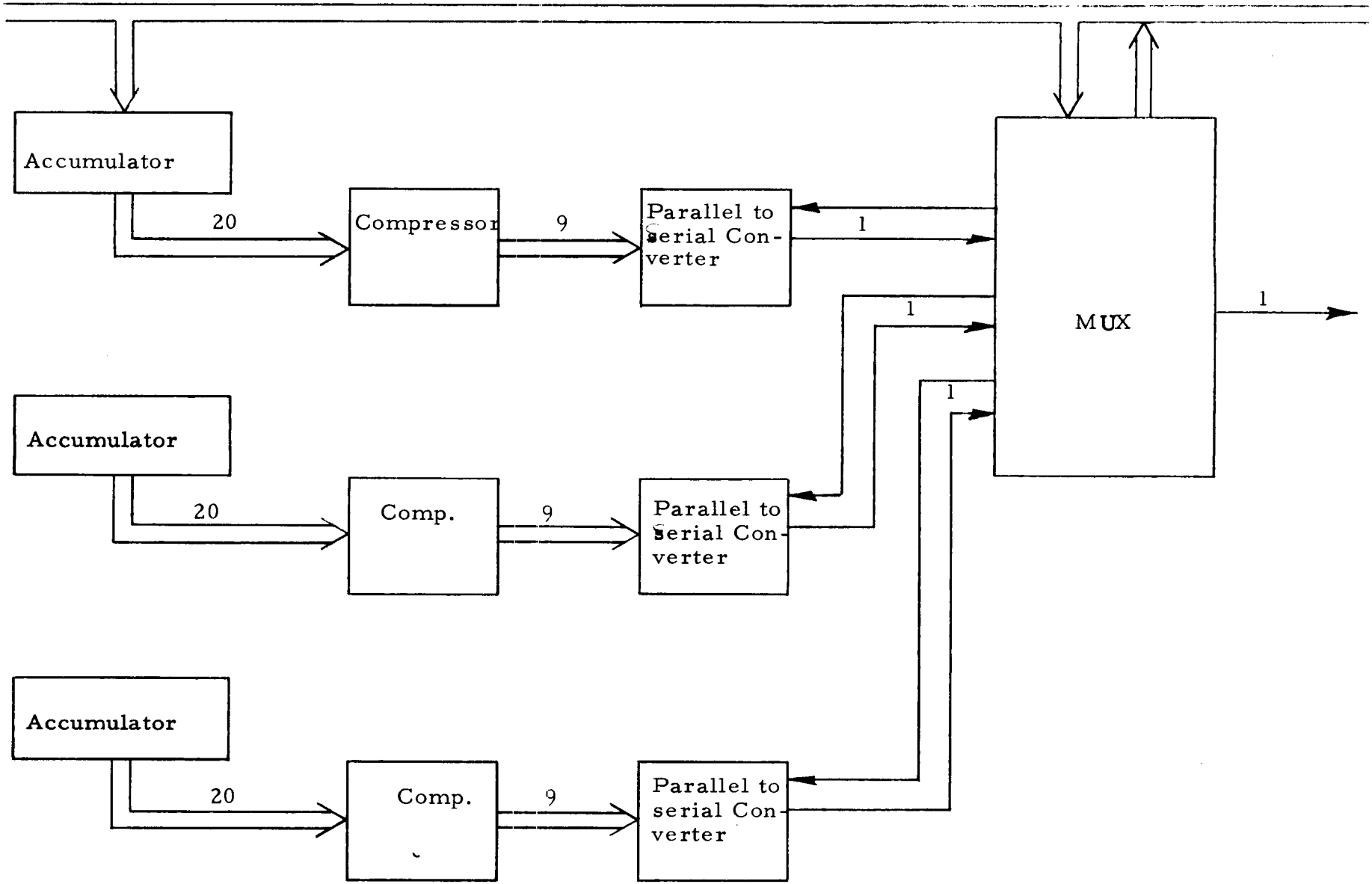
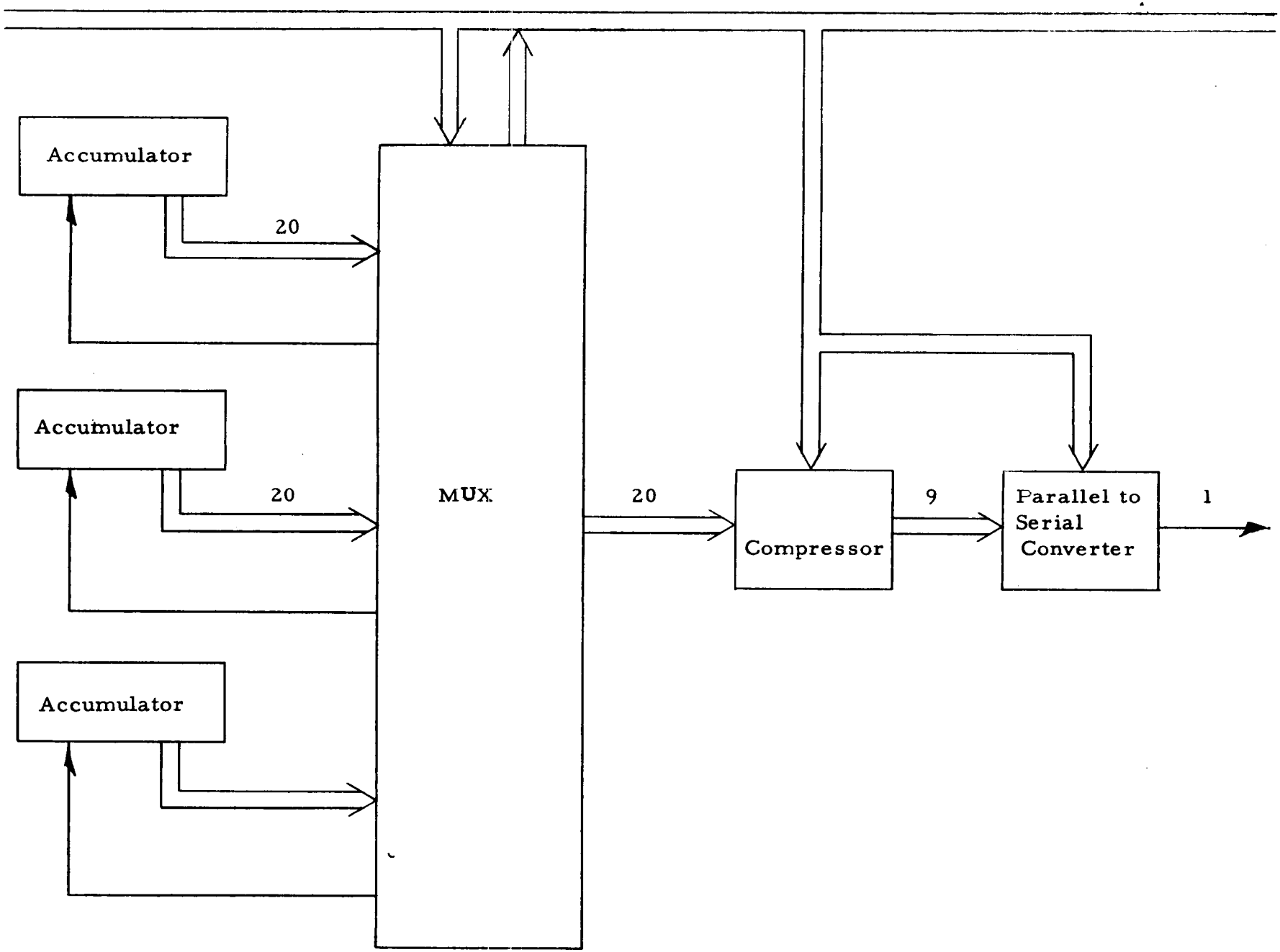


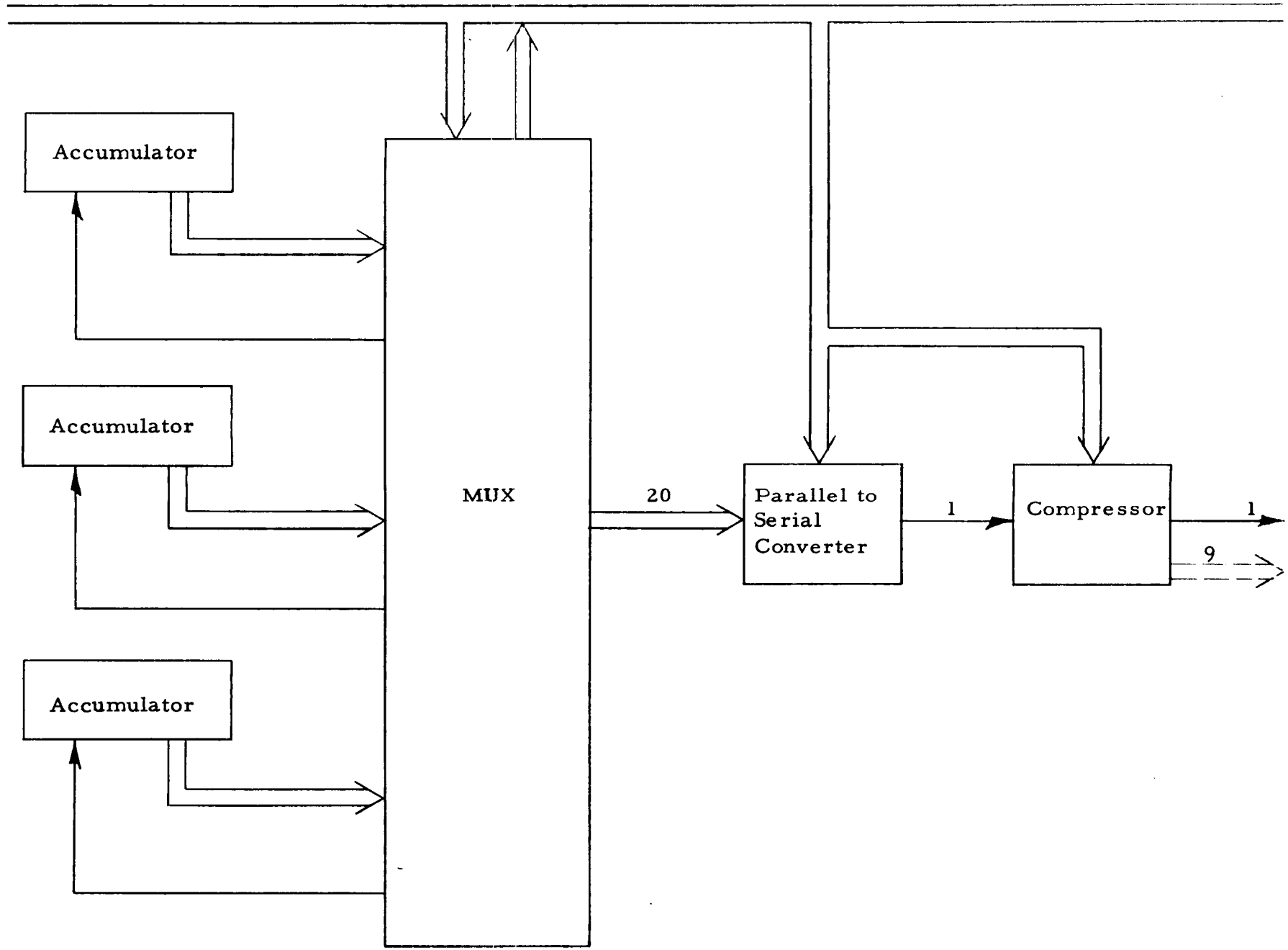
Figure 4.8-3

Multiplexer Configuration With Input Compression and Parallel/Serial Conversion

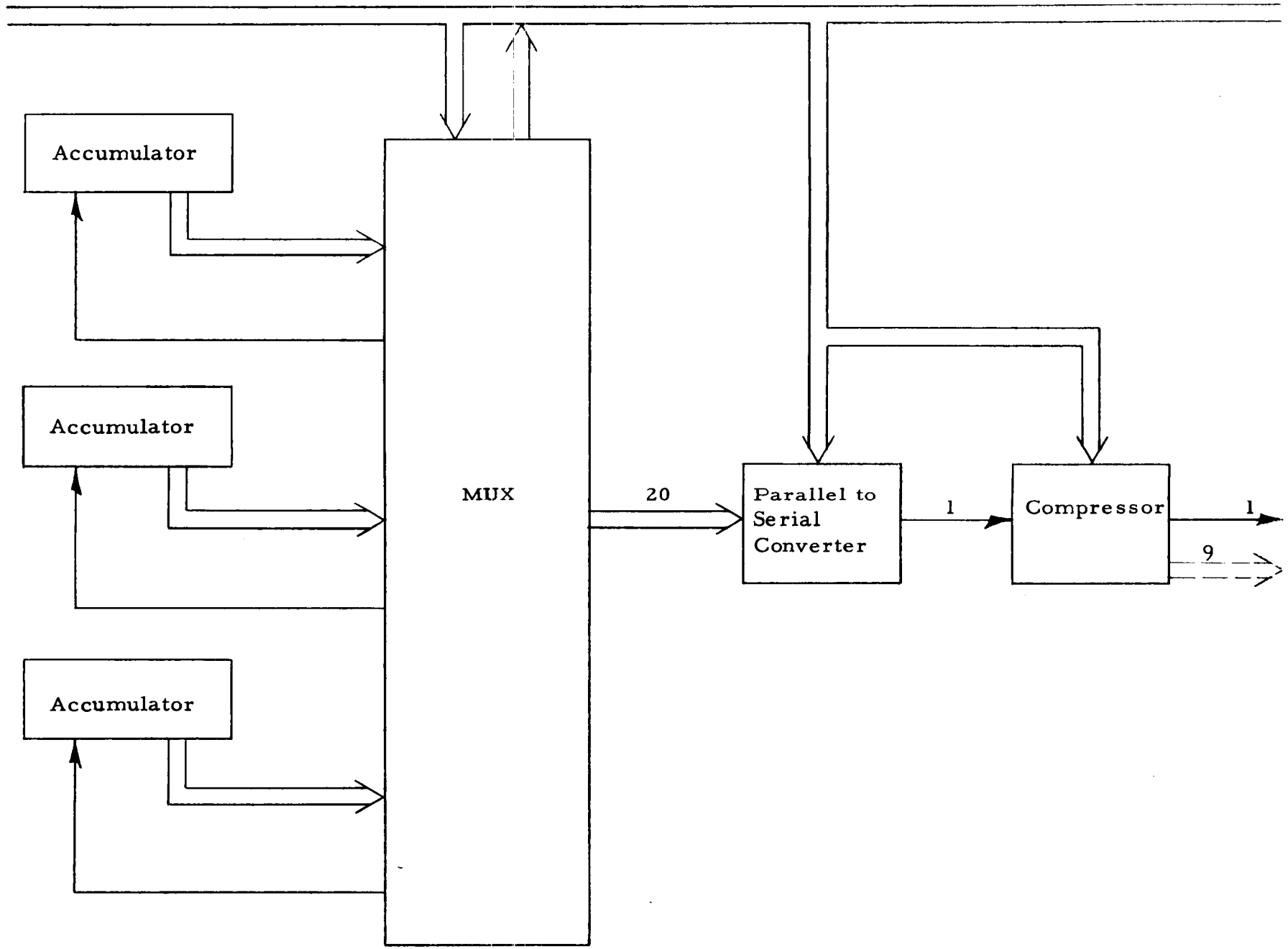


Multiplexer Configuration With Output Compression and Parallel/Serial Conversion

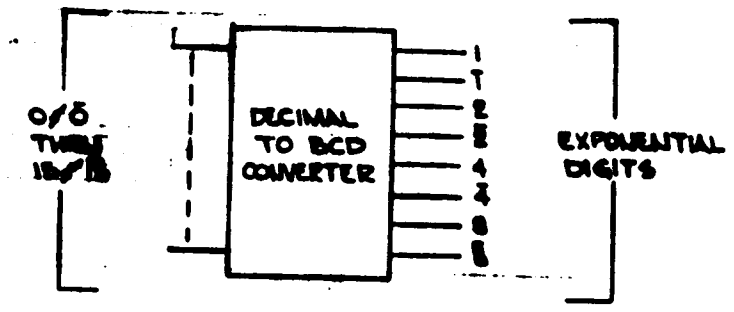
Figure 4.8-4

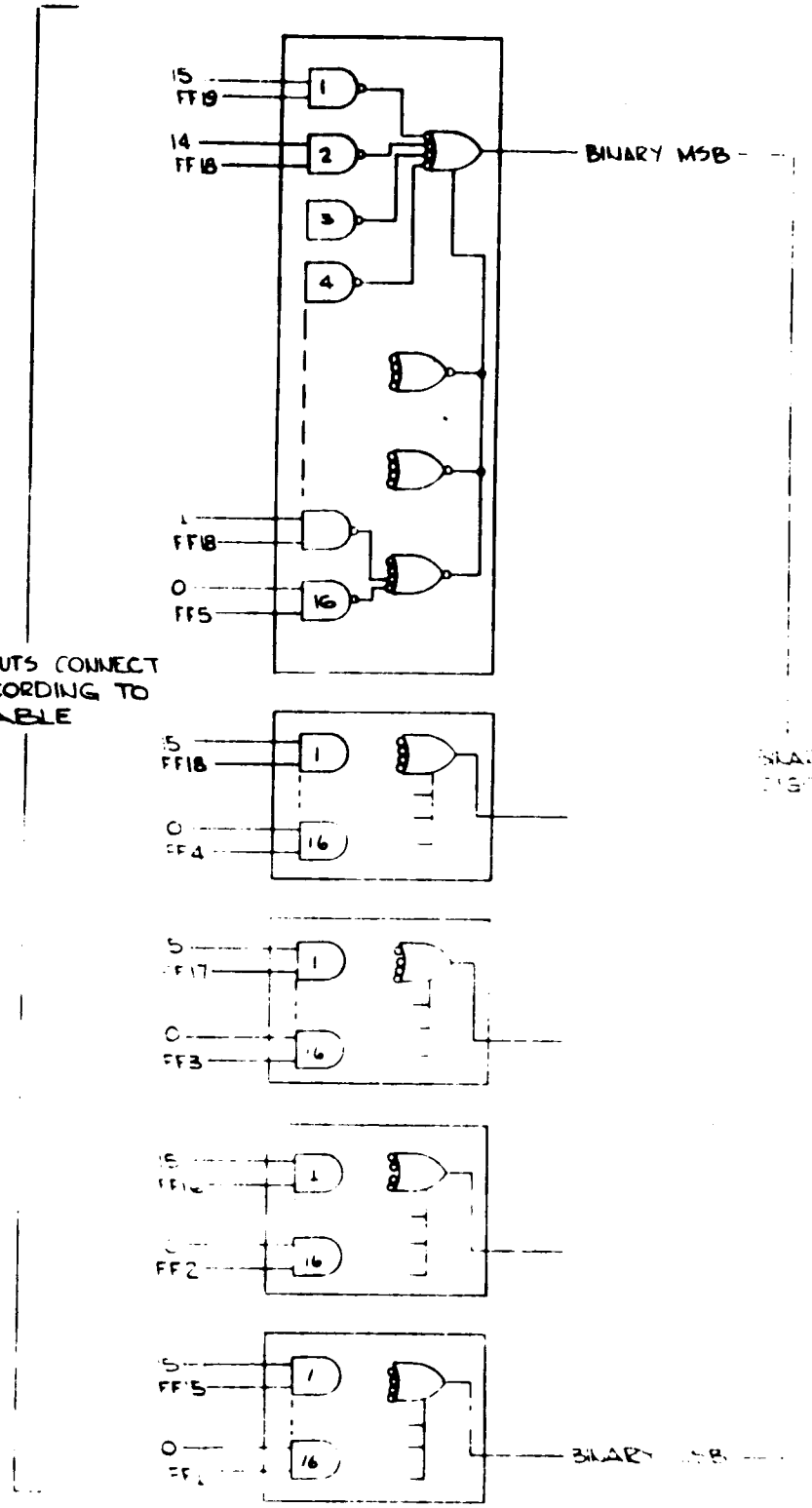


Multiplexer Configuration with Output Parallel/Serial Conversion and Compression
Figure 4.8-5



Multiplexer Configuration with Output Parallel/Serial Conversion and Compression
Figure 4. 8-5





STATIC COMPRESSOR
FIGURE 48-6

Using LPDT$_{\mu}$L, Figure 4.8-1 represents the best of the first three approaches and is better than the "log counter approach" discussed in the original proposal. A comparison of these approaches is as follows:

	Per Accumulator		Common Control	
Approach	LPDT$_{\mu}$L Packages	Power	Packages	Power
Figure 4.8-1	26	102 mw	71	31 6
Figure 4.8-2	47	182 mw	50	236
Figure 4.8-3	105	332 mw	50	236
Log Counter	40	160 mw	50	236

Table 4.8-1

Static Compressor Binary Digit Gating

Exponent	1 MSB	2	3	4	5 LSB
15	FF-19	18	17	16	15
14	18	17	16	15	14
13	17	16	15	14	13
12	16	15	14	13	12
11	15	14	13	12	11
10	14	13	12	11	10
9	13	12	11	10	9
8	12	11	10	9	8
7	11	10	9	8	7
6	10	9	8	7	6
5	9	8	7	6	5
4	8	7	6	5	4
3	7	6	5	4	3
2	6	5	4	3	2
1	5	4	3	2	1
0	5	4	3	2	1

4.8.2 Compression Techniques

For S^3, the data compression requirement is to reduce a 20-bit number to 9 representing the number to within 3% accuracy.

A common method used to represent a large number with a smaller number, is to let some of the digits represent exponential multipliers. For example; the normal binary number is of the form

$$S_1 = a_o + a_1 2^1 + a_2 2^2 + a_3 2^3 + ++ \ldots\ldots\ldots + a_n 2^n$$

however, by definition let us take a new number S_2 with 9 digits (X_o, X_1, X_2, X_3, X_4, A, B, C, D). For A or B or C or D not equal to zero

$$S_2 = a_o + a_1 2^1 + a_2 2^2 + a_3 2^3 + a_4 2^4 \qquad 2^{(A + 2B + 4C + 8D - 1)}$$

$$+ \; 2^{(A + 2B + 4C + 8D + 4)}$$

For A and B and C and D equal to zero

$$S_2 = a_o + a_1 2 + a_2 2^2 + a_3 2^3 + a_4 2^4$$

The coefficient values are shown in the following table, 4.8-2.

Table 4.8-2

Compression Coefficients

Coefficient Multipliers ($a_n 2^x$)					Exponential Multipliers				Exponential Value
a_o	a_1	a_2	a_3	a_4	A	B	C	D	
1	2	2^2	2^3	2^4	0	0	0	0	0
1	2	2^2	2^3	2^4	1	0	0	0	2^5
2	2^2	2^3	2^4	2^5	0	1	0	0	2^6
2^2	2^3	2^4	2^5	2^6	1	1	0	0	2^7
2^3	2^4	2^5	2^6	2^7	0	0	1	0	2^8
2^4	2^5	2^6	2^7	2^8	1	0	1	0	2^9
2^5	2^6	2^7	2^8	2^9	0	1	1	0	2^{10}
2^6	2^7	2^8	2^9	2^{10}	1	1	1	0	2^{11}
2^7	2^8	2^9	2^{10}	2^{11}	0	0	0	1	2^{12}
2^8	2^9	2^{10}	2^{11}	2^{12}	1	0	0	1	2^{13}
2^9	2^{10}	2^{11}	2^{12}	2^{13}	0	1	0	1	2^{14}
2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	1	1	0	1	2^{15}
2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	0	0	1	1	2^{16}
2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	1	0	1	1	2^{17}
2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	0	1	1	1	2^{18}
2^{14}	2^{15}	2^{16}	2^{17}	2^{18}	1	1	1	1	2^{19}

The largest number that may be represented is $2^{20} - 1$.

The definition of S_2, as given, satisfies two completely different approaches to the logarithmic data compression problem: logarithmic counter and sliding exponent.

Logarithmic Counter

A "Logarithmic Counter" which compresses as it counts is illustrated in Figure 4.8-7.

Flip-flops A through J are connected as a conventional 11-bit binary counter. Flip-flops M through T are also connected as a binary counter; however, the input may be inserted at any point in the counter chain. The gates that direct the input pulses are controlled by the high order bits of the 11-bit counter.

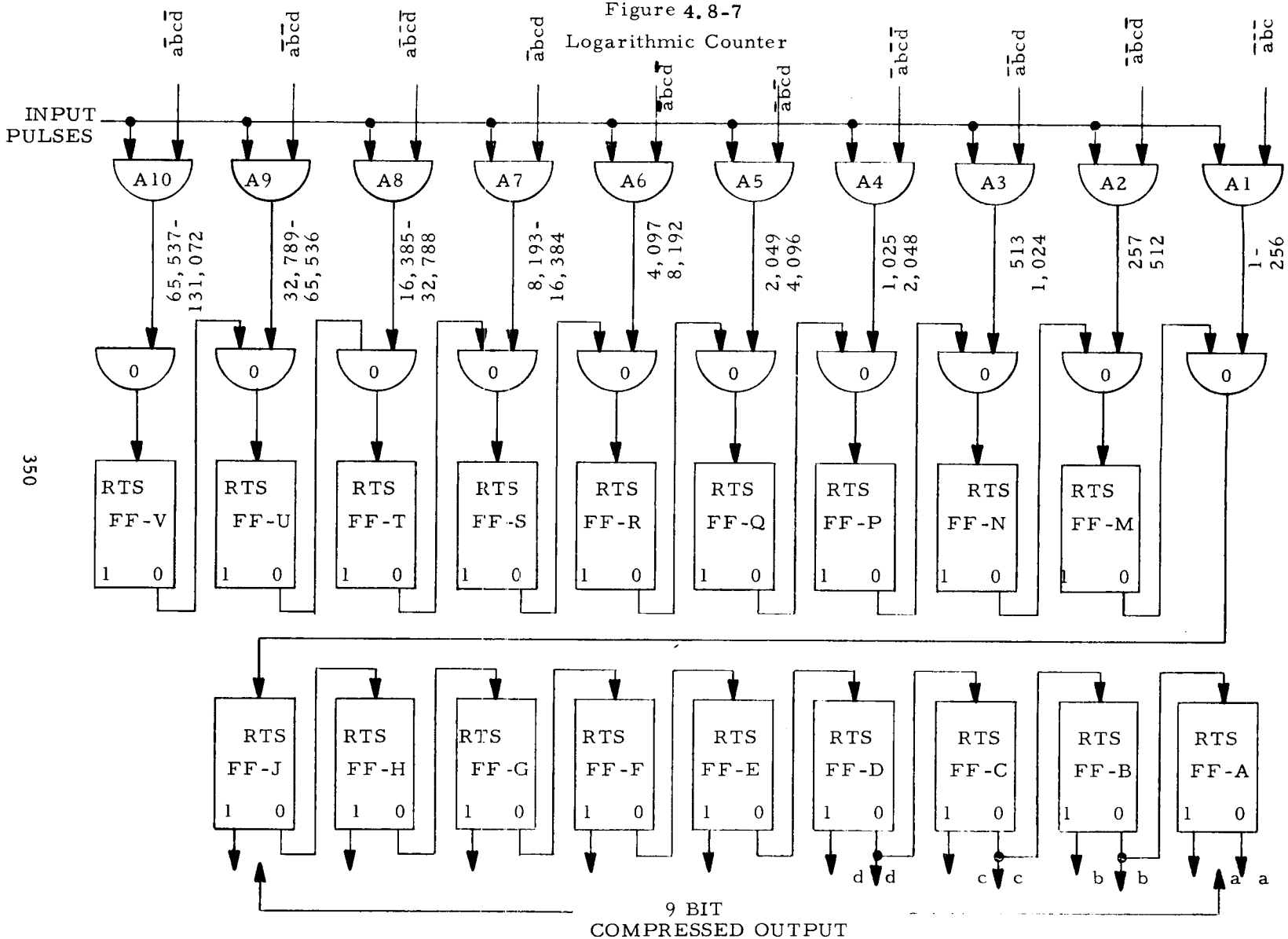
Initially, all flip-flops are reset and all gates, except A, are inhibited. The 256th pulse entered into the counter causes a change of state of flip-flop C, disabling A_1 and enabling A_2. Subsequent input pulses are directed to flip-flop M. As a result only every second input pulse is recorded by the 11-bit counter.

After the 512th input pulse, another stage of compression is introduced, allowing only every fourth input pulse to reach the counter input. As the total count increases, additional compression is introduced.

The principle disadvantage of this compression method is that the response times of the individual flip-flops and gates must be many times faster than adjacent input pulses. A logarithmic counter that is sufficiently fast to count pulses at up to 1 Mc. would therefore consume

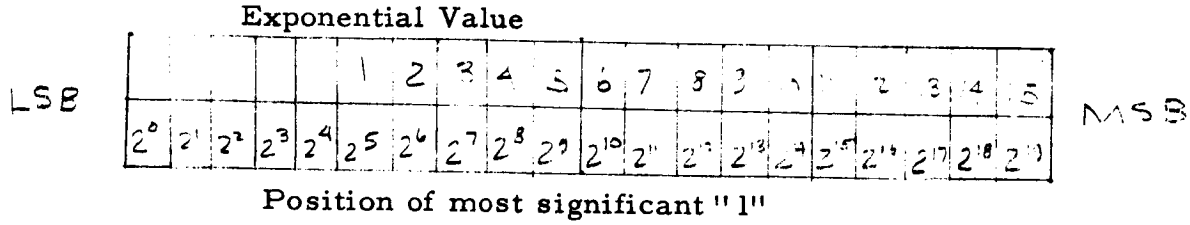
much more power than a conventional 1 Mc. binary counter, since faster logic is inherently more power consuming. The approach preferred is to accumulate the pulses in a binary counter and then shift the accumulated word into a "logarithmic" word compressor. The word compressor would be shared by each of the binary accumulators. One advantage of this approach is that slower logic can be employed. The second advantage is the reduction in total system complexity when several accumulators are used, because each accumulator is simply an ordinary binary counter.

Figure 4.8-7
Logarithmic Counter

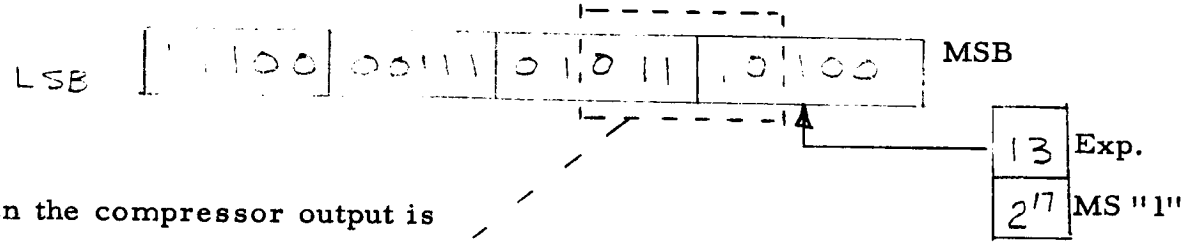


Sliding Exponent Compression

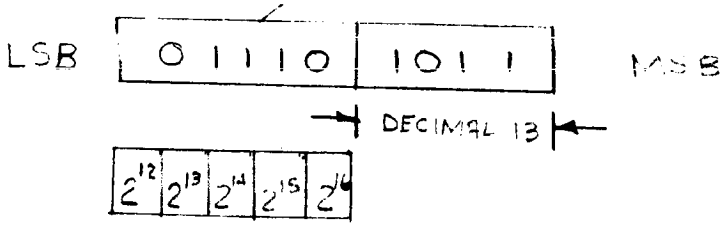
A dynamic compressor as described in section 3.0 is one in which the 20 bit binary number is examined to determine the location of the most significant "one" bit. The exponential value is assigned as shown



If, for example, the number to be compressed is



then the compressor output is



The remaining five bits are the next lower order bits after the first "one." If no "one" occurs in 2^5 through 2^{19}, the five bits are 2^0 through 2^4.

4.8.3 Compression Accuracy

The Compression Error for various combinations of exponents and binary bits is summarized in Table 4.8-3. The error is the same whether the sliding exponent or logarithmic compression technique is used. Note that the accuracy is always determined by the number of binary coefficients. The number of bits "n" that can be compressed by "e" exponential bits and "b" binary bits is given by

$$n = 2^e + b - 1$$

For example: e = 4, b = 5,

$$n = 2^4 + 5 - 1 = 16 + 5 - 1$$

$$n = 20$$

The maximum error is found in the following manner:

Case I Given a 20-bit number S_1, where

$$S_1 = 11\ldots11 = \sum_{N=0}^{N=19} 2^N = 2^{20} - 1$$

and S_2, the 9-bit compressed output, where

$$S_2 = 1111\ 1111 = (2^{14} + 2^{15} + 2^{16} + 2^{17} + 2^{18}) + (2^{19})$$

$$S_2 = \sum_{N=0}^{N=19} 2^N - \sum_{N=0}^{N=13} 2^N$$

$$\%\,\text{Error} = \frac{(100\%)(S_1 - S_2)}{S_1}$$

$$= \frac{(100\%) \sum_{N=0}^{N=13} 2^N}{\sum_{N=0}^{N=19} 2^N}$$

$$= \frac{(100\%)(2^{14} - 1)}{(2^{20} - 1)} \approx \frac{100\%}{2^6} = \frac{100\%}{64}$$

% Error = 1.56%

Case II Given a 20 bit number S_1, where

$$S_1 = 11\ldots11000001$$

$$= 2^{19} + \sum_{n=0}^{n=13} 2^n = 2^{19} + 2^{14} - 1$$

and S_2, the 9-bit compressed output where

S_2 = 00000 1111 = 2^{19}

$$\% \text{ Error} = \frac{(100\%)(S_1 - S_2)}{S_1}$$

$$= \frac{(100\%)\left[(2^{19} + 2^{14} - 1) - (2^{19})\right]}{2^{19} + 2^{14} - 1}$$

$$= \frac{(100\%)(2^{14} - 1)}{2^{19} + 2^{14} - 1} = \frac{100\%}{2^5 + 1}$$

$$= \frac{100\%}{33}$$

% Error = 3%

Case II gives the maximum error for the compressor used in the S^3 DPS.

Table 4.8-3

Compression Error

Data Word Length			Uncompressed Length	Maximum % Error
Total Bits	Exp.	Binary	Bits	
8	4	4	19	5.9
	3	5	12	3
9	4	5	20	3
10	5	5	36	3
	4	6	21	1.5
	3	7	14	.78
12	5	7	38	.78
	4	8	23	.39